CIRCULAR MANUFACTURING SYSTEMS FOR AEROSPACE APPLICATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Daniel Walton Cellucci

May 2019

CIRCULAR MANUFACTURING SYSTEMS FOR AEROSPACE APPLICATIONS

Daniel Walton Cellucci, Ph.D.

Cornell University 2019

Circular Manufacturing Systems are the foundation of a sustainable industrial ecosystem. These systems combine thorough instrumentation and near-perfect recyclability in order to produce objects whose primary existence is virtual, and whose physical instantiation is marked by informational support sufficiently extensive to provide robust insights into improvements to the original design. This dissertation explores three avenues of inquiry into the impacts of such a system to both aeronautics and space applications.

The first avenue concerns the design and implementation of a robot manufacturing system for space exploration, a printer capable of fabricating a wide variety of automata from a single strand of generic feed material. The second avenue concerns the development of a distributed instrumentation system for a prototype blended wing body aircraft and a novel edge computation algorithm for extracting high-level insights into aeronautic state like lift from low-level pressure data. The third avenue concerns the discovery of a novel class of open-cell cellular solids, derived from Triply Periodic Minimal Surfaces. This avenue presents analysis of these lattices using two different methods, a direct stiffness approach and symmetry-extended counting rules, and shows that the D-Schwarz open-cell lattice is the first known example of a three-dimensional Tensegrity Lattice Material, a class of lattices that have both a collapse mode and a prestressable self-stress state that is capable of stiffening the structure.

BIOGRAPHICAL SKETCH

Daniel Cellucci was born in Los Angeles, California, to Mary Jonelle Cellucci and William "Willy" Cellucci. He was raised in Atlanta, Georgia, attending Chamblee Charter High School, and wanted to become a physicist since his first year when he attended an afterschool program at Fernbank Science Center.

Daniel attended the University of Georgia majoring in Physics, but took many more sculpture classes than was reasonable or necessary after a chance encounter in a drawing class his freshman year introduced him to Sculpture Professor R.G. Brown. With Professor Brown, he learned how to cast glass, how to bake bread, and how to make a decent dovetail joint. He also worked closely with Michael Oliveri from the Art-X department, who taught him how to run a CNC router and introduced him to electronics like the Arduino. In physics, he studied ab-initio electronic-structure calculations for SnO_2 nanocrystals under Dr. Steven P. Lewis.

After graduating from the University of Georgia, Daniel realized that applying his scientist training in the world of sculpture was not that much different from engineering, and so, in 2012, he joined the Cornell University Department of Mechanical and Aerospace Engineering working under Dr. Hod Lipson. As a NASA Space Technology Research Fellow, Daniel attended NASA Ames Research Center to work with Dr. Kenneth Cheung on reversibly assembled cellular composite materials. Shortly after arriving at NASA Ames, he switched advisors to Mason Peck, and stayed at Ames until his graduation in 2019. In 2018, he joined Ames as Pathways Intern in the Applied Manufacturing Division, a hybrid civil servant/student position. He is currently a co-Primary Investigator on a project involving distributed sensing in aircraft, and a researcher on the Distributed Satellite Autonomy project.

This thesis is dedicated to my father, Willy Cellucci.

He used to say, ”Just knock it out, it’ll take you five minutes if you try”.

Here we are, seven years later...

ACKNOWLEDGEMENTS

I would like to begin by thanking my advisor, Dr. Mason Peck, without whom this dissertation would have been abandoned.

To my mother, Mary Jonelle Cellucci, thank you for always being there to help me through the rough patches.

To Nick Cramer and Sean Swei, thank you for being the most thoughtful, insightful, and capable collaborators anyone could ask for.

To Alex Mazhari, Christine Gregg, Ingrid Desilvestre, Marcia Sawyer, and Robert MacCurdy, thank you for your patience and understanding.

To Martin Miller, Caroline O'Donnell, Giffen Ott, Michael Oliveri, and R.G. Brown, thank you for making my life and research more interesting.

To the New York Public Libraries, the Multnomah County Libraries, and the San Francisco Public Libraries, thank you for providing the space to complete this thesis.

Lastly, and most importantly, thank you Michelle Fong. You know why.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation is an exploration of the engineering behind making a SPIME.

What is a SPIME? Also known as a SPace-tIME object, it is a design concept originally introduced by Bruce Sterling [102]. A SPIME spends most of its life as uninstantiated data on a server somewhere else. When it is needed, a machine manufactures a copy of it, embeds it with sensors, and provides it to the user who requested it. The SPIME then lives its life and logs every moment of its use. When it is no longer needed, the user disposes of it. Instead of the SPIME going to a landfill, however, the materials are decomposed into their original constituent parts, ready to be assembled into the next object, and the collected data are sent back to the server hosting the design. This server then combs through the data for insights that can be used to improve the design. The next time the SPIME is requested, it is manufactured with these improvements. For example, consider a pair of SPIME shoes. These shoes first appear as a Computer Aided Design (CAD) model in an interface. They will be customizable: color, texture, lace type, et cetera are all options. The one option that is not present is the size— the user instead uses a CAD model of each of their feet that, when their design is ready, a printer uses to make a pair of shoes that fit better than any standard size could. The opportunities and challenges of the manufacturing infrastructure required to produce such a customized shoe have been well described, usually in the context of additive manufacturing [67]. Sterling himself refers to these endlessly customizable objects as GIZMOs, a sort of predecessor to the SPIME that focuses on extensibility at the cost of stability. However, there

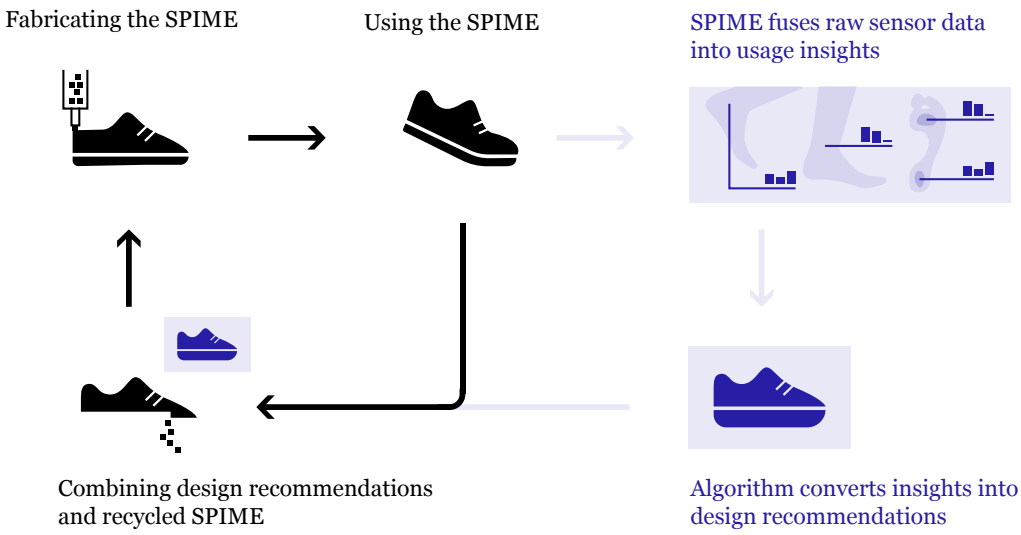

Figure 1.1: A Pair of Spime Shoes (figure designed by Michelle Fong)

are two critical capabilities that make a SPIME distinct from a GIZMO: the data that the shoes collect and the ability to seamlessly recycle the shoes back into the matter stream once their use has been completed. The motivation for the former capability is functional, the latter is existential.

With the former capability, the data collected during the SPIME's lifetime completes the picture for a GIZMO by providing the peripheral information that is not available without testing in the operating environment. Perhaps the user tends to walk on the balls of their feet, so the shoe's sole wears there first. A SPIME shoe notices this, and the next shoe will have harder material in these worn spots. Perhaps the user just moved to a town like Ithaca that likes to salt every available surface when it snows. A SPIME shoe learns this, and the next shoe uses a material that is resistant to salt. These kinds of externalities might be predictable with a robust enough model, but with seamless manufacturing and recycling, it is easier to build the object, use it, and derive the lessons learned

from the results than it is to try to predict all of these scenarios beforehand.

With the latter capability, design for recycling is the basis of Sterling's thesis on SPIMEs, and the Viridian Design Movement [101] for which Shaping Things was one of many manifestos. Sterling was not just concerned with the SPIME as an engineering challenge, Sterling wanted to imagine the kind of society that would be built on SPIMEs. Foremost, he theorized, a SPIME society would be concerned with its future, and would try to act in a way that maximized its longevity. From this perspective, a manufacturing system that pollutes the Earth, produces large amounts of CO_2, and sends the vast majority of its products to a slow death in a landfill is an existential threat that must be addressed. Sterling's theory is not so much that this society is inevitable, but that this society is, through some kind of natural selection, the only one guaranteed to survive such a self-imposed threat. That is, either modern society adopts the SPIME view or it consumes itself.

The critical insight for Sterling is we, as humans, already recycle and already iterate. We do these things poorly, slowly, and at great expense. Getting an improved pair of shoes is an intellectual labor that the user performs by sifting through the avalanche of products that companies produce in order to meet every possible demand as quickly as it is expressed. Since we are categorically bad at cradle-to-grave manufacturing [58], the vast majority of these products end up, unused, in the landfill.

SPIMEs were proposed to solve both of these problems simultaneously. They will be adopted, according to Sterling's and the Viridian Design Movement's thesis, because they save the user and the producer time, money, and effort. SPIMEs will save the user effort because the product is a distillation of the

user's needs and desires into a single, customizable, optimal design, instead of a vast range of static, half-useful products that need to be laboriously sought out, tried on, and agonized over before becoming a part of the user's life. SPIMEs will save the producer effort because they will no longer have to deal with stock or sales or clearance racks. Most importantly, however, both the user and the producer will make a self-interested decision based on convenience, and, in doing so, will unintentionally save the planet from being buried under a layer of not-quite-fashionable trash.

However, the reality is not so frictionless as Sterling suggests— it is not simply sufficient to suggest this system to the existing economic order and expect it to be adopted wholesale because it is rationally better. In addition to the variety of social impediments, there are numerous technologies that must first be developed in order to enable such a means of production. These technologies are necessary for the SPIME ecosystem but not exclusive to it— they can be employed in a variety of ways, one of which may be conducive to creating SPIMEs.

This dissertation begins with the assumption that the aerospace industry is the ideal place for the development of the technologies that enable SPIMEs. In particular, SPIMEs meet two critical needs of the aerospace industry. The first need concerns increased operational efficiency. Billions of dollars are spent every year to improve existing aircraft designs— performance increases of even a few percent can save millions of liters of fuel and thousands of metric tons of CO_2. The data that SPIMES provide into the details of their use can form a critical component of aircraft design, providing the sort of insights that resulted in scalloped nacelles and wingtips, without the massive investments in research infrastructure required to manually discover these improvement. The second

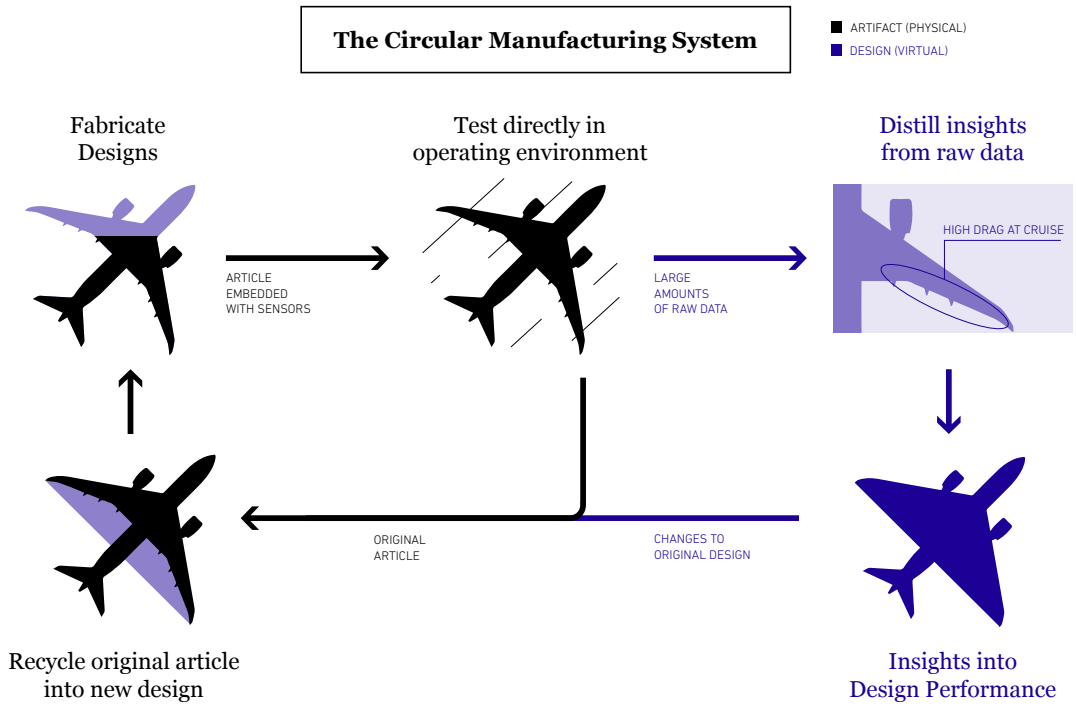

Figure 1.2: The Circular Manufacturing System for an Aircraft (figure designed by Michelle Fong)

need concerns material reuse and reconfigurability. When a plane is decomissioned, it sits in a graveyard in the Mojave until it can be manually disassembled. Multi-stage spacecraft discard the expended stages, leaving them to either drift for eons or (more optimistically) fall to Earth, burning up in the process. The materials that compose these artifacts are usually the state of the art when they are built, representing the most advanced, energy-intensive, and expensive components of their time. However, when the artifact's useful lifetime has been completed, in aerospace, it often is not or cannot be disassembled.

What does an aircraft or spacecraft SPIME look like? What technologies are necessary to make these SPIMEs a reality? These are the questions that this dissertation attempts to answer, through three works that each explore a part of the

larger problem. The vehicle through which this dissertation explores these questions is an approach called a *circular manufacturing system*. A *circular manufacturing system* provides the infrastructure possible to realize the essential qualities of a SPIME. It contains both the ability to produce an object and disassemble this object. It also has the ability to augment the object with the sensors necessary to measure the performance of the object being produced, and read this information at the end of the object's life. This system can then comb these data for the kinds of insights necessary to improve the object's design. Figure 1.2 shows an example *circular manufacturing system* for an aircraft.

The first work explores a basic *circular manufacturing system* for planetary exploration SPIMEs. It describes a machine that can produce an automaton tailored to a specific task from a strand of feed material, then recycle that automaton back into itself and produce a second automaton optimized for a different task. In total, the machine produces three different automata from this single strand of material, each capable of locomoting in a distinct manner. While this system demonstrates a rudimentary form of reusability, it is essentially an open-loop system— the robots are not instrumented in a way that allows for performance to be measured in an automated manner, and there is no means for these insights to be automatically incorporated into the design algorithms. The closing section of this work is devoted first to a successor design of a ribosomal robotics system incorporating lessons learned from this first version, and then to describing a set of experiments that can demonstrate data collection and incorporation into the design model.

The second work tackles the problem of "closing the loop" on a *circular manufacturing system* by demonstrating that lightweight embedded sensor networks

can provide realtime estimates of critical aeronautics parameters like lift. Such sensor networks can provide *in-situ* estimates of the flight envelope and true operating conditions of future aircraft, shortening the design process and increasing fuel efficiency. These estimates can act as the inputs into design algorithms that can then suggest improvements to the object being manufactured in order to improve operational performance. While these design algorithms are left for future work, the closing section of this chapter focuses on the possibilities of SPIMEcraft, aircraft that embrace this approach and how they might accomplish many of the goals set out in various policy documents and roadmaps for commercial aviation.

Finally, the third work examines the material problem of SPIMEs by studying structures that can be assembled and disassembled easily. It introduces a new class of cellular solids that display high-performance with low connectivity and locally flat connections. These Reversibly Assembled Cellular Solids can provide clear weight savings in a variety of high-performance applications, while remaining repairable and replaceable[18]. The cellular solids in this work provide a high level of performance, while also providing the affordances necessary for automated assembly.

1.1 Contributions

The six major contributions of this dissertation follow three sub-themes:

1. Robotic autonomous manufacturing, which includes the first two contributions:

   (a) Designing and building a novel robot manufacturing system that can

produce several specialized automata using a single strand of feed material.

(b) Demonstrating the first robot manufacturing system capable of producing designs from recycled material.

2. Distributed sensing and algorithms, which includes the next two contributions:

   (a) Designing a novel distributed instrumentation system for aircraft, which shows that low-cost, low-footprint Micro-Electrical-Mechanical Systems (MEMS) pressure sensors can accurately estimate aerodynamic state.

   (b) Developing a novel algorithm for performing the distributed calculation of aircraft lift from pressure values.

3. Advanced Cellular Materials and Structures, which includes the final two contributions:

   (a) Discovering a new class of open-cell periodic lattices derived from Triply Periodic Minimal Surfaces.

   (b) Analyzing Triply Periodic Minimal Surface (TPMS) lattices and performing experiments indicating that the lattice derived from the D-Schwarz surface is the first known three-dimensional Tensegrity Lattice Material.

Together, these six contributions provide the foundation for the technologies required to build a true SPIME system. While it does not demonstrate a fully-integrated *circular manufacturing system*, it does show the pieces of the system and provides a vision for how these pieces can be combined.

CHAPTER 2

RIBOSOMAL ROBOTICS

This work describes an end-to-end manufacturing system that bends a length of feed material augmented with motors into an automaton that can then locomote. This approach to manufacturing allows an automaton to be recycled back into the system when its task is completed by unbending the material. As a piece of the circular manufacturing system, Ribosomal Robotics explores the process of fabricating, testing, and then recycling designs. Figure 2.1 highlights this part of the cycle by surrounding it with a red box. This work, however, does not explicitly produce an article embedded with sensors, though the motor modules are capable of housing such sensors.

My contributions include:

1. designing and building the wire bender and the feed material including the motor modules.
2. writing the software necessary to communicate with the bender.
3. performing the experiments and recording the results.
4. analyzing the scalability of the process by examining the space of possible materials that could compose the wire.

However, I claim no credit for the development of design algorithm, that should go to Sebastian Risi. I only provided feedback for the design algorithms in order to ensure the resulting designs were manufacturable. I am also not responsible for development of the firmware or selection of the control boards for the motor modules, credit for that should go to Rob MacCurdy.

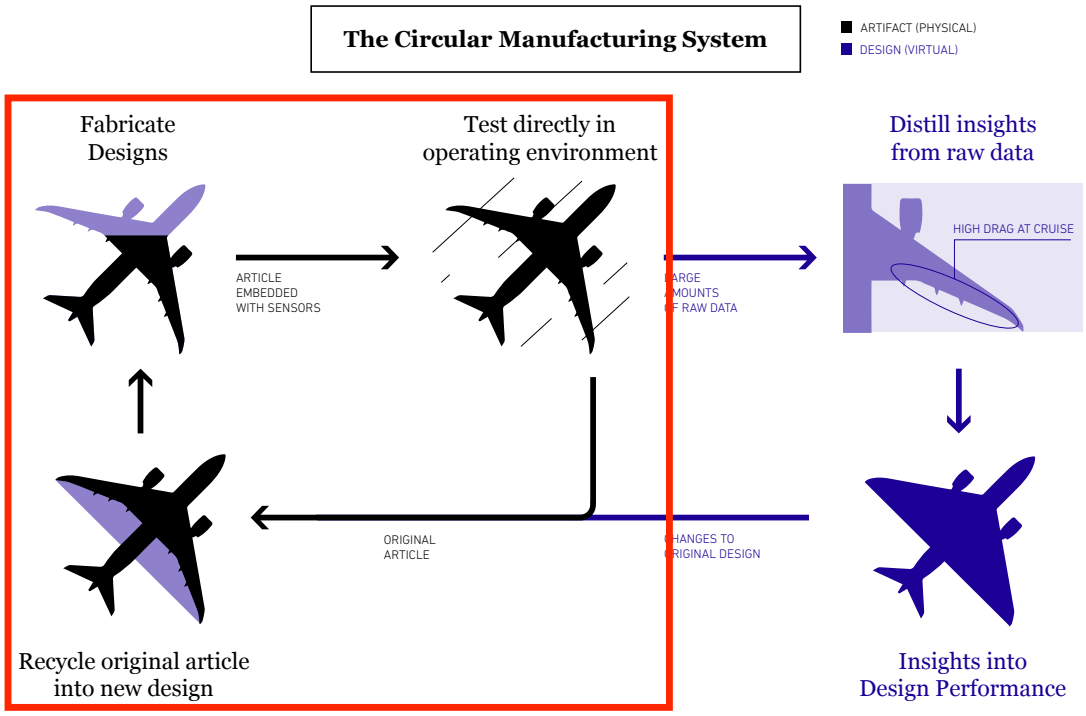

Figure 2.1: The portion of the circular manufacturing system that Riboso-mal Robotics Explores

2.1 Background

While advances in 3D printing have allowed robot mechanisms to be produced with greater ease and speed [71], and new additive manufacturing materials and processes are beginning to enable on-demand printed circuits [72, 107, 110], the 3D printing of complete systems that include actuation and energy storage is still in its infancy. The potential of this design and manufacturing scheme has not yet been leveraged to fabricate complete robots; they are still manually designed and constructed, a complex, time-consuming process that requires ex-

perts at all stages. The major goal of a robot "walking out of the printer" is not realizable with current additive manufacturing technology. Additionally, utilizing *recycled material* is infeasible with virtually all 3D printing methods. Though in principle a 3D-printed object made from a single material could be reduced to its base material and reused, the facilities required to perform this operation make it impractical for robot applications in remote and inaccessible locations (where an on-demand and reusable robot fabrication system would be particularly useful). Also, this recycling approach is not applicable to robots fabricated from multiple materials.

To address these challenges, this chapter describes a proof-of-concept, fully-automated design and assembly process, inspired by the ribosome, which can automatically discover solutions to high-level design challenges and instantiate the designs as physical artifacts. The contributions of this work are as follows:

1. outline a one-dimensional robot fabrication concept.
2. demonstrate the technical aspects required to implement the printer and its source material.
3. describe the theoretical details of the optimization process used to design the robots.

Previous work, (e.g. GOLEM) have shown how evolutionary methods could design simple moving robots that were then partially fabricated automatically [68]. Other researcher have advanced the robot design process by allowing a human designer to compose modular subsystems [78, 22], or shown how specific pre-designed robots can self-fold [38]. This work demonstrates a complete, autonomous system that synthesizes designs from high-level behavior specifi-

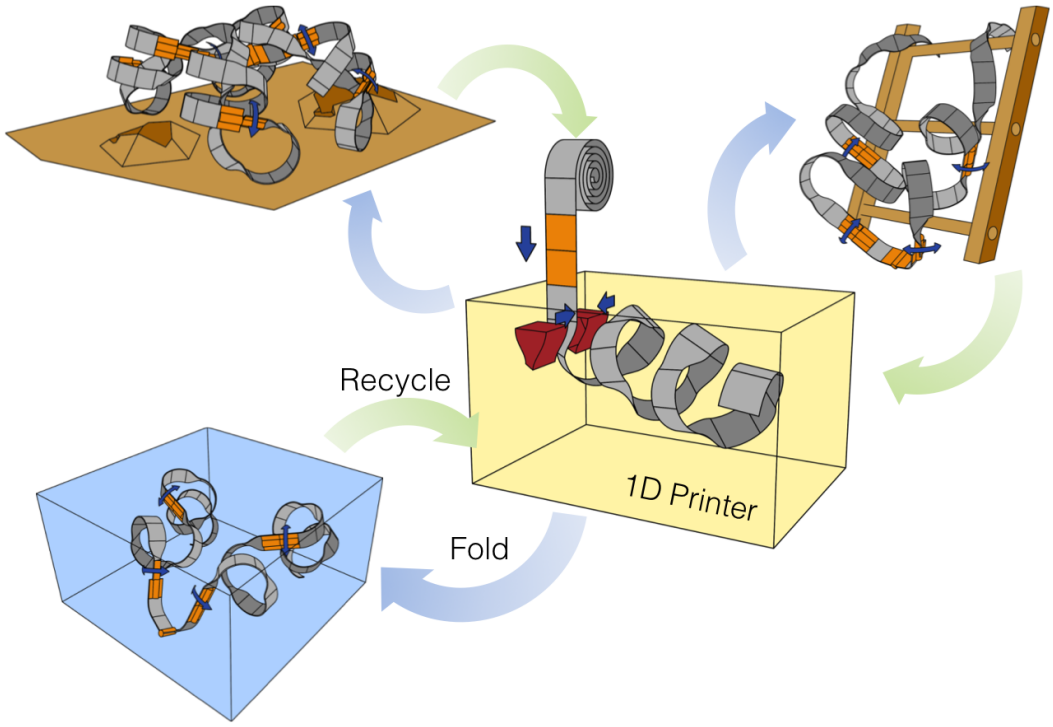

Figure 2.2: One-Dimensional Robot Manufacturing System

cations and then automatically fabricates ready-to-use robots. This autonomous system employs an evolutionary-based approach to discover the sequence of folds required to create a specialized automaton from a one-dimensional strip of material. Actuators and other control elements augment the *pre-manufactured* source material and a custom printer folds this material into the prescribed configuration Figure 2.2 shows this robot manufacturing system. The one-dimensional printer (center) folds specialized robots for different tasks on-demand, all derived from a single generic strip of deformable source material (shown in gray) with *pre-embedded* actuators and control elements (shown in orange). The system can reuse the *same* material to fold a different three-dimensional robot for a different task by unfolding the deformable parts of the

material and feeding the strip back into the printer. This allows the printer to recycle these robots into a different design when no longer needed. The ability to recycle material forms the basis of a generalized method for automatically creating robots tailored to a particular task. This chapter describes how this system can automatically design and fabricate robots for three different locomotion tasks, and how a robot designed for one task can be recycled into another.

Previous examples of folding applied to robotics include strings that self-configure into complex structures passively through magnets [46] or via electrical motors [16, 61], origami-inspired systems that generate 3-dimensional robots from two-dimensional planes with actuated hinges [38, 50, 80], 3D-printed objects that fold in response to temperature [73] or humidity changes [90], and machines that can manufacture non-actuated mechanisms from a flat ribbon [108] or filament [65]. Printing integrated electromechanical systems that include sensing, computation, actuation and energy storage is a persistent challenge for approaches that build with raw materials like plastic filament or conductive paste. Robot systems based on prefabricated modular designs sidestep these challenges [43, 70, 82], though they share one or more common drawbacks, including relatively large module size, high complexity and cost, as well as module-interconnect challenges. The complexity, strength, and cost of electromechanical connections between modules has been specifically identified as an ongoing issue [83, 33], and the approach described here was chosen to avoid these problems. Specifically, the angle-holding torque of an individual module does not determine the overall size and power consumption of the resulting robots (a challenge noted in [61]).

This work extends previous efforts to automatically fabricate robots by re-

laxing the requirement that the robot *self-reconfigure*, placing that capability in a dedicated assembler instead. Doing so removes complexity (and associated energy, cost and size implications) from the fabricated robots. In contrast to self-folding approaches that either require dedicated hardware at every possible fold-site [16, 61, 50] (regardless of whether any particular design uses the fold or not), or are designed for one particular robot [38, 80], our approach relies on the reversible deformation of a material (metal wire) that is low-cost, readily mass-produced, and can create a multitude of designs with no human effort. Brodbeck et al. [13] developed a similar system, where a robot arm assembles modular robots. However, the one-dimensional printing approach allows the production of customized, application-specific geometries with fewer discrete modules and, consequently, with less structural complexity and cost.

The approach taken by biological systems during protein synthesis inspires the use of an external fabrication apparatus to impart a particular desired structure onto a generic input material. This apparatus, called the ribosome, enables the construction of myriad chemicals that form the basis of all cells through the ordered sequential assembly of amino acids [40]. In particular, the ribosome also plays a role in determining the ultimate morphology of a protein as it is assembled by modulating the synthesis rate, which impacts the folding pattern [97]. The ribosome is clearly a distinct entity from the approach explored here, with many other characteristics not embodied in the manufacturing system's design. However, adopting the use of an external folding mechanism allows this system to convert a simple, generic, linear input material into a variety of special-purpose robots.

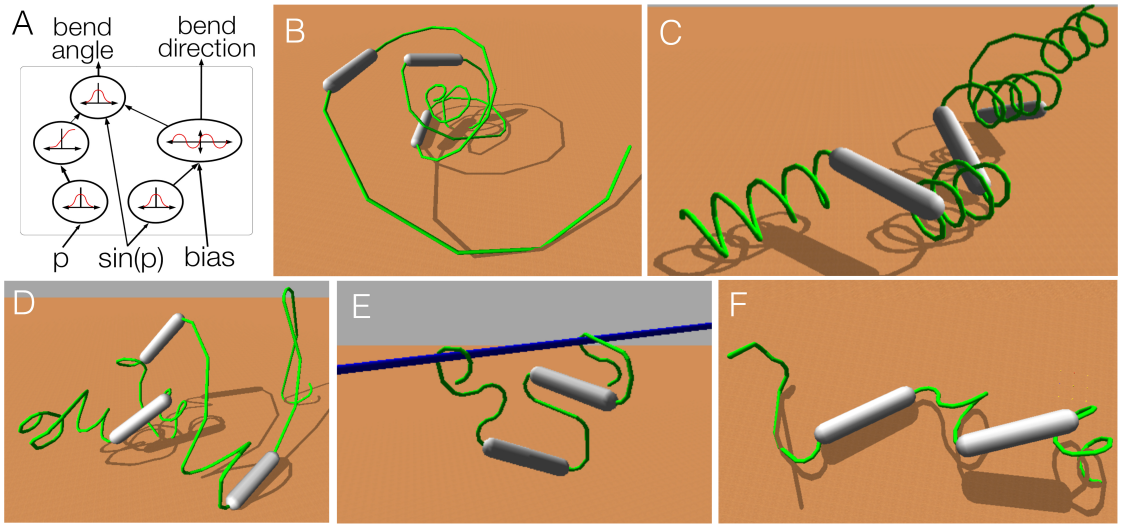

Figure 2.3: The Compositional Pattern Producing Network (CPPN) used to encode the designs, and some example designs derived from these encodings.

2.2 Generating an Optimal Design

The robot manufacturing system creates robots by sequentially folding a 1-dimensional wire until it produces the desired 3-dimensional robot. The wire contains pre-embedded actuators at fixed intervals, allowing different segments of the wire to rotate relative to each other. These actuators communicate wirelessly, allowing coordinated motion control. Each motor module can apply a rotation in the interval [-90, 90] degrees between the two wires that are connected to either end of it. An optimization algorithm generates robot designs and tests them in a simulation environment. In order to reach an optimal design in a reasonable time, the algorithm needs a way to represent the design of the robot that is both robust, compact, and capable of great variation. This section describes the method of representing the robot designs and the optimization algorithm that produces a robot.

2.2.1 Design Representation

A modified version of a *compositional pattern producing network*(CPPN; [98]), a special type of artificial neural network, encodes the pattern of folds in the wire. CPPNs draw their inspiration from evolutionary development, and can create complex artifacts such as two-dimensional images [95], three-dimensional forms [20], and connectivity patterns of high-dimensional neural networks [41]. While they have been used previously to encode morphologies of simulated robots [92, 8], this work demonstrates, for the first time, the transfer of CPPN-encoded robots to the real world. The key concept behind CPPNs is that they generate a solution to a problem by iteratively composing more primitive functions in a directed graph, adding functions and weighted connections to the graph until they reach a satisfying solution. The resulting functional representation is *generative*: it does not require as many internal parameters as the morphology of the object that it defines would dictate. Imposing this structure on the representation of the object dramatically reduces the dimensionality of the search, making large, complex problems tractable.

In this work, the CPPN for each robot generates the sequence of folds required to describe the robot's morphology by applying itself along the length of the robot's body (Fig. 2.3A). The encoding of this CPPN utilizes activation functions with regularities such as as symmetry (e.g. Gaussian) and repetition (e.g. sine) to facilitate the discovery of robot designs that satisfy the desired behavior specifications (Fig. 2.3B-F). Importantly, CPPNs also allow regularities with variation [98], which is challenging for more regular indirect encodings such as L-systems [52].

The inputs to the CPPNs (Fig. 2.3A) are the current segment number p scaled

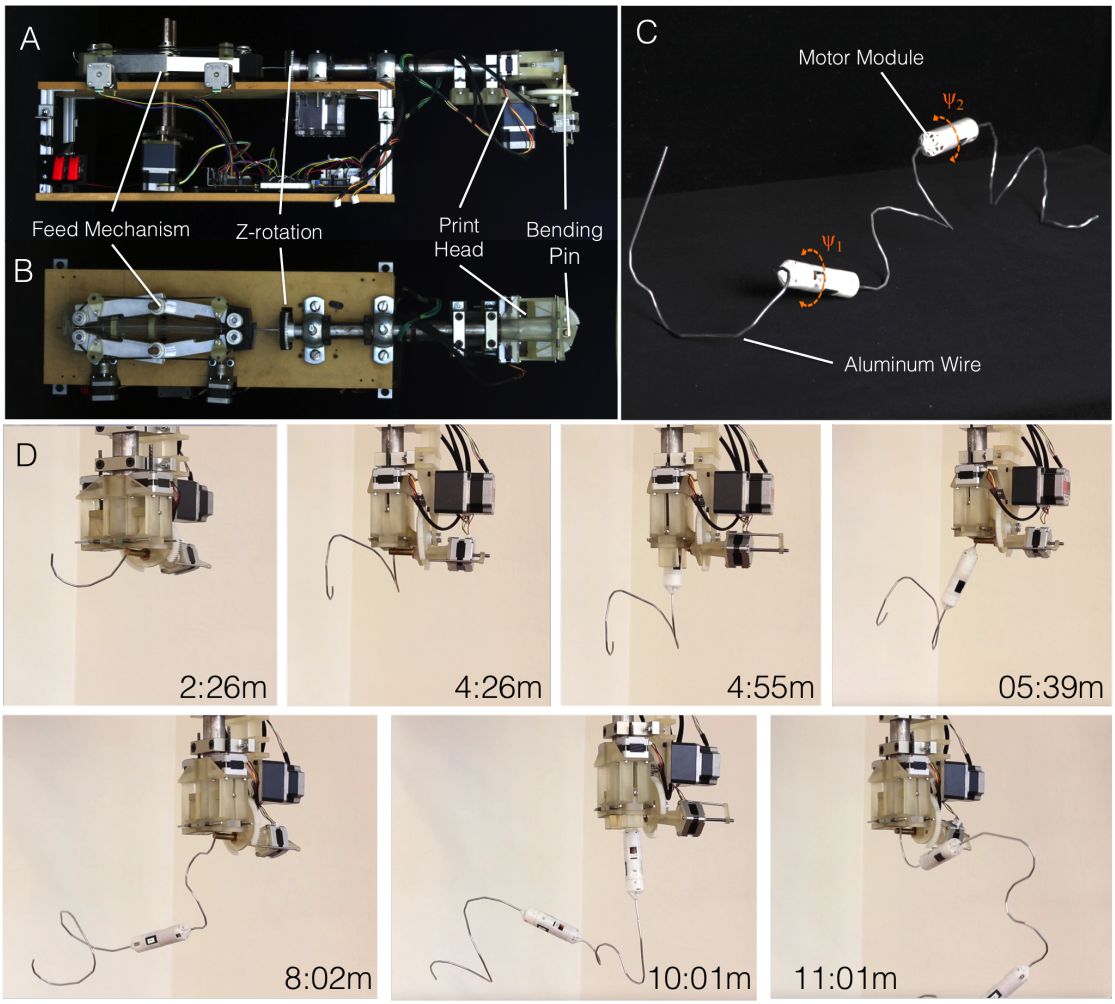

Figure 2.4: The printer, the process of printing, and an automaton produced using this printer.

to [-1,1], and $(sin(p) + 1)/2$, which facilitates the evolution of structures with repeating patterns. The z-rotation output (*bend angle*) determines the rotation of the bend head, while the second output b (*bend direction*) determines the direction of the fold (+30 degrees if $b < 0.0$, -30 degrees otherwise). If a fold results in a collision of the wire with the printer, then the design skips the fold and the wire instead simply advances forward (i.e. the printer feeds but does not fold) one segment length. In addition to the morphological description, two CPPN

outputs (not shown in Fig. 2.3A) encode the motor control signals, automating the motor controller-design task. These two outputs determine the amplitude A and phase φ for a modified sine wave motor-activation function: $A\ sin(t + \varphi)$. For each motor with position p on the string, the CPPN encodes a value at that location that determines the specific amplitude and phase values that control the angle of each motor module. The CPPN scales the value of φ to $[-\pi/2, +\pi/2]$.

2.3 Optimization

The optimization process employs a multi-objective evolutionary computation approach [25] to produce a set of CPPN-encoded robots in a 3D rigid-body physics simulation using the freely-available Open Dynamics Engine. Controlled tests provide the necessary calibrations to the simulation parameters (e.g. friction, maximum motor torques and speeds, material density), in order to minimize the difference in the behavior of the robots between simulation and reality. Evolutionary algorithms have shown promise in solving complex engineering tasks with multiple competing objectives and large numbers of decision variables [34]. They have also contributed successfully to different robot design tasks [11], motivating their application here. The potential design space of a robot composed of N individual segments, each of which can be bent either up or down by 30° in 180 different orientations, is $2^{N-1}180^{N-1}$. The typical robot designs explored in this work assign $N \in [50...150]$, rendering an exhaustive brute-force search infeasible (the number of possible configurations exceeds the estimated computational capacity of the universe [69]). In contrast, the search space explored by the CPPN-based approach increases gradually during evolution, and the complexity of the final representation does not typically exceed 50

connections. For example, the CPPN encoding the automaton in Fig. 2.3E has 30 connections and the automaton in Fig. 2.3F has 50 connections.

The *Neuroevolution of Augmenting Topologies* (NEAT) algorithm [99, 100] optimizes the CPPNs for the specific tasks required of the resulting robot. This algorithm can evolve neural networks and therefore also CPPNs. The initial population in NEAT consists of random CPPNs, in which the network's inputs are directly connected to its outputs. NEAT then adds connections and nodes over the course of evolution, making them more complex. The size of the network does not need to be set a priori; because NEAT grows candidate solution networks, it avoids unnecessarily searching through high-dimensional solution spaces when a simpler solution is adequate.

Three objectives determine the fitness of each robot:

1. maximizing its speed in the specified domain
2. maximizing the *compactness* of the produced design
3. minimizing the bending-torque required during folding

The process iteratively selects machines with greater fitness, creating offspring by modifying the underlying CPPN description via mutation and crossover between more fit individuals in the population. To encourage the evolution of a diverse population of designs, this algorithm employs the popular multi-objective optimization approach NSGA-II [25], together with *novelty search* [64]. Novelty search offers a more exploratory and divergent evolutionary search than traditional objective-based methods by augmenting the fitness function with a novelty metric that rewards diverse phenotypes. The equation below describes the

novelty p of an individual x in morphological space,

$$\rho(x) = \frac{1}{k} \sum_{i=1}^{k} \text{dist}(x, \mu_i), \tag{2.1}$$

where μ_i is the ith-nearest neighbor of x with respect to a distance metric $dist$. The metric $dist$ is the average euclidean distance between the vectors of folds that describe different robots. If the novelty is above a threshold ρ_{min}, then the individual enters into a permanent archive. The novelty algorithm examines only the k-nearest neighbors in the joint set of individuals in the archive and the current population for computing the novelty of an individual. In this work, $k = 15$.

In addition to novelty, the second objective in the NSGA-II approach is a traditional fitness function that rewards individuals for traveling as far as possible in the allotted amount of time: $T = |p_s - p_e| \, (1.0 - tq) \, c$, where p_s is the starting position of the robot and p_e is the ending position after the evaluation period. To facilitate the evolution of designs that are within the design space of the printer, the fitness function tries to minimize tq, which is the maximum torque on the design during the folding process, and maximize c, which the compactness of the design (measured as: 1.0 - average distance over all segments to the center of mass of the robot). While it is possible to avoid some collisions during the folding process by skipping a particular fold, collisions with the printer can also occur while the printer is simply feeding material (e.g. if a part of the robot bends around the back of the printhead and then the printer feeds it forward). Therefore, the algorithm rewards design designs for also not colliding with the

printer by adjusting the multi-objective fitness:

$$F_1 = \frac{T}{1 + cl}, F_2 = p, \tag{2.2}$$

where cl is the number of collisions during the simulated folding process.

While each of the three navigation tasks (crawling, pipe traversal, and rolling) could have a custom fitness function, as is typical in evolutionary computation [34], this work shows that only scaling the CPPN outputs (Section 2.2) to slightly different ranges is necessary for the evolution of high-performing solutions. For the walker and roller, A falls between [0, π], while for the pipe-traverser, it falls between [0, 1] in order to prevent the robot from swinging violently and potentially falling off the pipe (since this possibility was not modeled during simulation). Additionally, there are hard limits on the overall range of the motor movement. The crawler and pipe-traverser motors must remain within the range ±45 degrees. For the roller, both motors follow an identical sine function (are determined by querying the CPPN for the only first motor position provides the values of A and φ for both motors), and the motors must remain within the range ±90 degrees. These constraints encourage the evolution of designs that locomote by rolling.

Optimization Parameters. The size of each population is 100 with 10% elitism. The number of generations is set to 300. Sexual offspring (75%) do not undergo mutation. For asexual offspring (25%), the probabilities of link weight mutation, link addition, and node addition are 0.75, 0.1 and 0.05, respectively. The available CPPN activation functions are sigmoid, Gaussian, absolute value, cosine, and sine, all with equal probability of being added. Parameter settings are

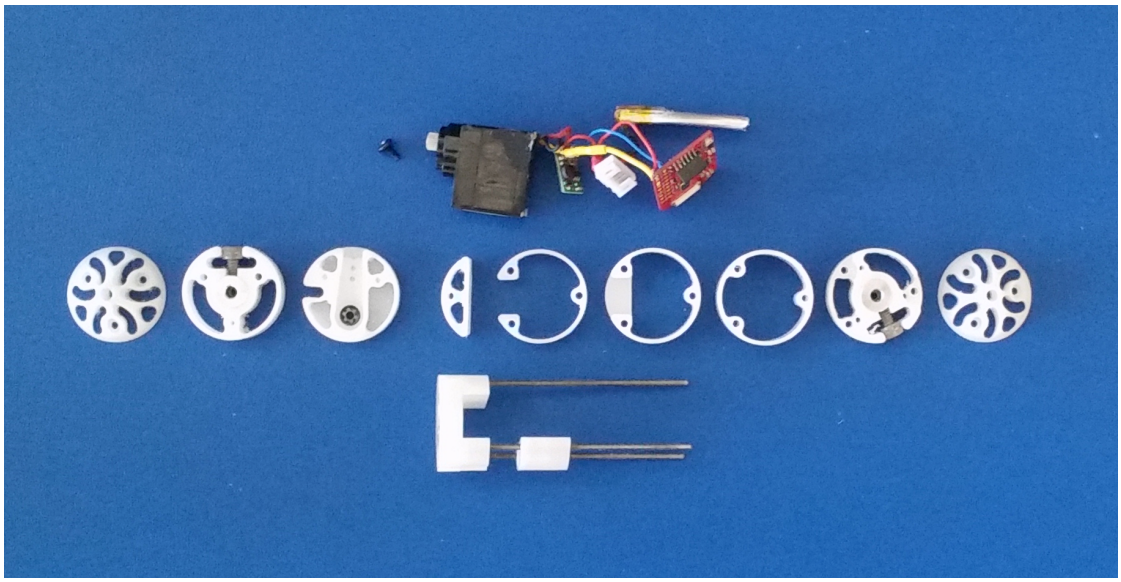

Figure 2.5: A Deconstructed Motor Module

consistent with prior reported settings for NEAT [99, 100].

2.4 Robot Manufacturing System

The input material for our printer is 3 mm diameter 1100-alloy aluminum wire with motor modules embedded at regular intervals along its length. The wire exhibits a soft (bend-and-stay) temper, which minimizes bending error due to elastic versus plastic deformation. Each motor module contains a Hitec HS-5065MG+ servo, a 100 mAh lithium-polymer battery, a voltage converter, and an MSP430-RF2500T wireless microcontroller. An acetal Delrin homopolymer housing encloses these components and connects to the two adjacent aluminum wires via shaft collars at each end. This housing is a cylinder that is 26 mm in diameter, with conical endcaps and an overall length of 110 mm. The modules use aluminum shaft collars embedded in the case to attach to the aluminum wire.

The total mass of each motor module, including batteries and components, is 62 g.

Figure 2.5 shows one such module deconstructed, in order to illustrate the parts that compose it. The white parts are the milled delrin pieces that form the structure of the module, which consists of a two halves. The front half uses the three leftmost parts, and contains a conical end cap, a part that holds the shaft collar, and a part that holds the servo horn. Three bolts hold these pieces together. The second half uses the remaining parts. It consists of a part that holds the servo, three layers of spacers that provide the volume to house the electronics, another shaft collar mount, and another endcap. The servo mount also holds the threaded rods, which provide the alignment necessary to ensure that all these parts stay together. One of the spacers illustrates this how these parts use these rods for alignment. Additionally, there is a small window that provides the slot for plugging in either a charger for the battery or a jumper that connects the battery to the rest of the electronics. Finally, there is a notch on both the horn and servo sides of the motor mount that allows a single alignment pin. This pin ensures that the mount does rotate while the machine bends the wire, and is removed before testing.

The aluminum wires that mechanically connect adjacent motor modules do not provide any electrical connectivity. Since the clocks in each module are independent, a synchronization mechanism coordinates the movements of the motors. The wireless microcontrollers in each motor module provide this synchronization via a leader-follower scheme: one motor module provides the clock for all the other modules. The leader periodically broadcasts a message that includes the absolute time. When a follower module hears this message, it sets its

local clock and continues the motor command playback sequence. Each motor module has its own motor sequence, a function that maps time into motor position, which the evolved CPPN determines, as described in Section 2.2. These parameters arrive to the motor module along with the start command, in the form of a wireless transmission over a 2.4GHz ISM-band signal.

The D.I.Wire Bender (www.instructables.com/id/DIWire-Bender/) provides the initial reference design of the machine that folds the material into the target morphology (Fig. 2.4A,B). This machine requires three additional mechanisms that allow it to accommodate the larger-diameter motor modules. The first mechanism is a wire-feed system with two sets of arms that can either close around the wire to grasp it, or open to let a motor module pass through. The second is a print-head with a sliding door through which the material passes. The door can open to allow a motor module to pass, or close to grip the wire. Finally, the printhead also has a rotational degree of freedom, allowing it to rotate around the material's feed axis (Z-rotation).

2.4.1 Feed Mechanism

The feed mechanism consists of two sets of identical arms with knurled steel drive cylinders mounted on the ends. The drive cylinders grip the wire tightly when the arms closed, and a single stepper motor rotates them using a set of gears and timing belts. Two additional stepper motors drive the opening and closing of the front and back set of arms. These steppers connect to the arm mechanisms with a custom turnbuckle that feeds into two threaded brass blocks, which then bolt onto each of the arms. The steppers drive the turn-

buckle from one side, instead of the middle as is typical. A flexible shaft coupling provides the tolerance necessary to allow for the lateral movement of the turnbuckle as the arms open and close. Finally, 3D-printed alignment flanges fitted on the end of the arms recenter the wire onto the drive cylinders in the event of a misalignment.

The length of the arms is sufficiently large to allow a motor mount to fit between the drive cylinders, allowing this mount to pass through the mechanism by opening and closing the arms in a specific sequence, as illustrated in Figure 2.6. Both arms begin engaged, as shown in frame A. When the motor mount approaches the rear arms, these arms open, producing the configuration in frame B, and the front arms feed the material until the motor mount passes the rear mechanism. After the mount passes the rear mechanism, the rear arms reengage the wire, producing the configuration in frame C. The front arms then open, producing the configuration in frame D, and the rear arms feed the material into the pipe that leads to the print head. Once the motor module passes into this pipe, the mechanism returns to the configuration shown in frame A.

2.4.2 Printhead

The printhead consists of a spring-loaded door mechanism driven by two threaded stepper motors. This door provides the clamping force during the bending step, while still opening wide enough to admit a motor module. The door mechanism and the printhead consist of a set of 3d printed components that slide against one another. The printhead attaches to the feed cylinder with a friction fit, while the other components use teflon tape and graphite dry lu-

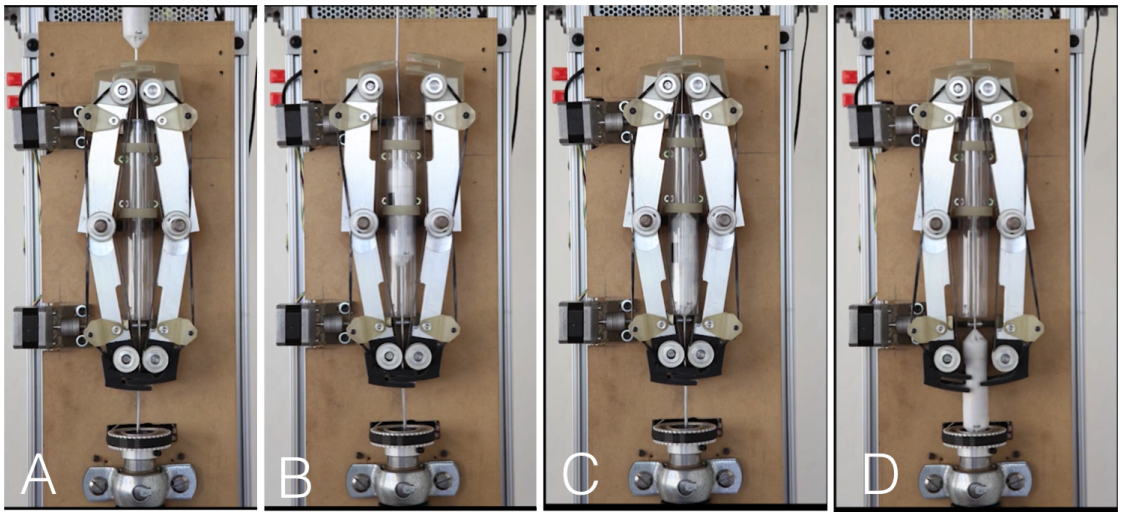

Figure 2.6: Sequence of operations that allow the printer to feed a motor module through the feed mechanism.

bricant in order to slide without stalling. A pair of stepper motors drives the doors open and closed by moving a pair of inclined plane blocks forward and backward. These blocks contact the doors at a 45 degree angle, transferring the forward/backward motion into opening/closing motion. Two springs provide the restoring force necessary to reopen the doors as the blocks move into the forward position.

Attached to the underside of the printhead is a stepper motor that drives the bending pin (a brass cylinder) into contact with the deformable wire. Additionally, a lead-screw driven by a stepper motor and attached to the the bending pin moves the pin up and down, allowing it to retract far enough to allow a motor module to pass or to move under the wire to perform bends in the opposite direction. Figure 2.7 shows the process for feeding a motor module through the print head. When the module approaches the doors of the print heads (Frame A), the two steppers motors on the side of the head drive down the blocks, and

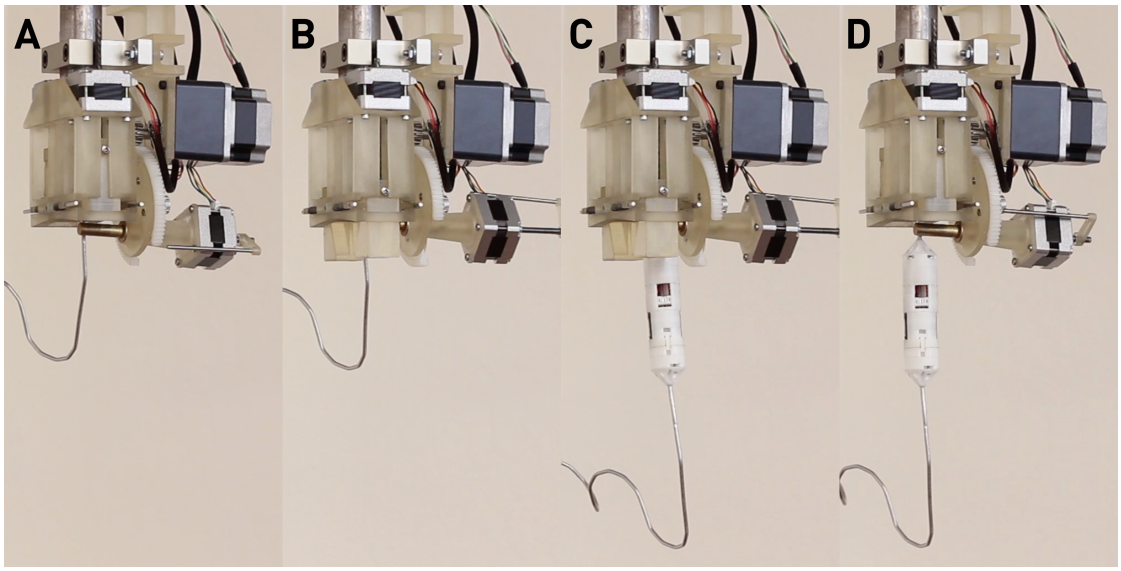

Figure 2.7: Sequence of operations that allow the printhead to open around a motor module.

the stepper holding the pin lowers it out of the path of the module (Frame B). The printer then feeds the length of the module though this opening (Frame C), and then recloses the doors and raises the pin (Frame D). In order to reduce the chance of the pin and the motor from interfering, the printer feeds the motor well past the pin and then pulls the module back toward the printhead.

In order to ensure that the printed robots accurately represent the simulated designs, calibration of the printer involves optimizing the printer commands to perform ±30 degree bends. The error in the resulting bend angle at 30 degrees is 2 degrees with a standard deviation of 0.84 degrees. The largest source of error is bend-back in the wire, in which the elastic deformation of the material produces a bend a few degrees smaller than desired. The largest source of variation is due to irregularities in the linearity of the unbent wire. Bending the wire 2 degrees past the target angle reduced the error due to bendback, and ensuring that the feed material was as straight as possible reduced the variation due to

irregularities.

2.4.3 Z-rotation

The Z-rotation degree of freedom allows the printhead to rotate around the feed axis, enabling bends to occur in any plane parallel to and intersecting this axis. This degree of freedom permits the formation of complex three dimensional structures from a succession of two-dimensional bends. The circular cross-section of the wire simplifies the bending process, since the same contact surface occurs regardless of z-rotation. However, this cross-section also allows the wire to rotate slowly within the machine while it feeds through. As a result, if there is sufficient moment applied from the bent part of the wire, then the printer can lose track of the θ-position (the wire slowly rotates as it is fed). Mounting the printer vertically mitigates this moment by aligning the feed direction with the gravity vector. This reorientation of the printer reduces the impact of gravity on the drift in the θ-position, but limits the maximum length of the resulting robot. Section 2.5.1 provides some estimates of the maximum length due to this effect.

2.5 Printing Examples

The printer (Fig. 2.4) produced ten complete robots in the course of development and testing; four worked as designed. Initial failure cases were due to mismatches between simulation and reality, as well as miscalibrations of the printer's bend angles. The input material for the examples shown here all have an overall length of 2 meters (87 segments) with two motor modules embedded

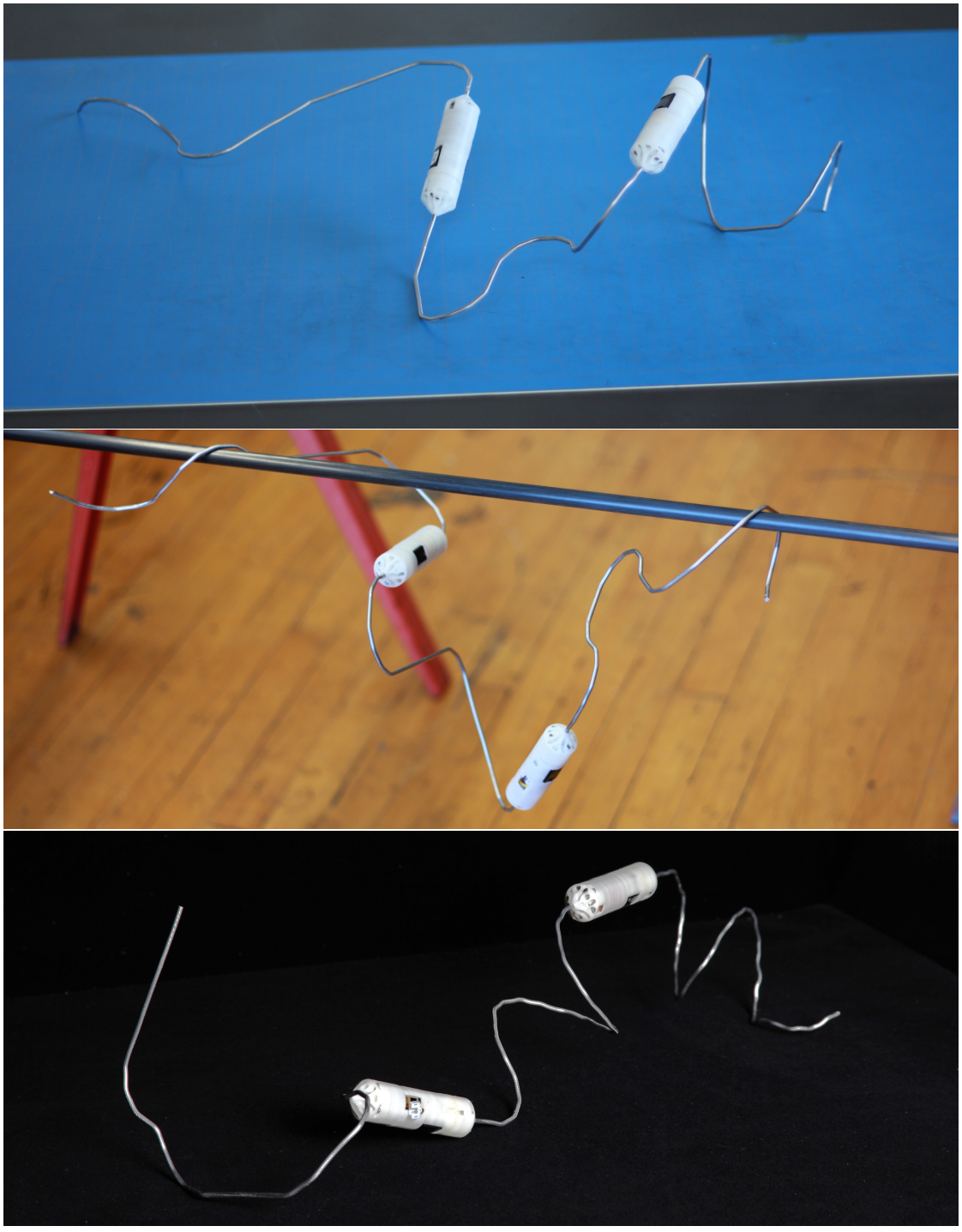

Figure 2.8: The three produced designs (from top to bottom): walker, pipe traverser, and roller.

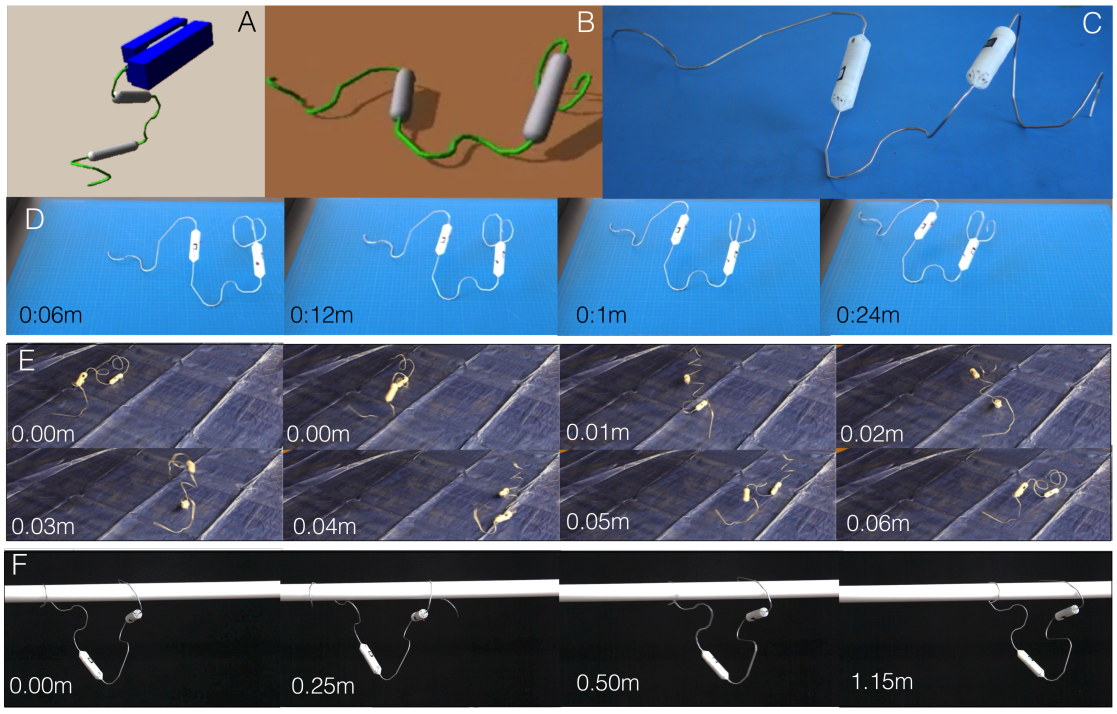

Figure 2.9: Experimental results for the three tested scenarios.

at 0.76 m intervals (Fig. 2.4C) that subdivide the wire into three equal-length sections. Folding an 87-segment design takes approximately 13 minutes (Fig. 2.4D). The only manual step involves removing the completed robot from the printer and executing the evolved controller on the motor modules in order to actuate the robot.

Figure 2.9 shows three printed physical robots in action. The crawler robot (Fig. 2.9D) moves by using its two motors to rock its center of gravity forward and backward. On the forward cycle the rear portion of the robot loses contact with the ground, allowing its motor to move it forward. On the rearward cycle the rear portion pushes off the ground and the cycle repeats. The pipe traverser (Fig. 2.9F) employs a related strategy; by fully encircling the pipe, the rear section of the design allows the robot to grip the pipe, which allows the front

Table 2.1: The number of motor cycles each robot required to travel its full length.

	Cycles/Body Length Traveled	
Morphology	**Virtual**	**Physical**
Crawler (Fig. 2.9B, 2.9D)	29	32
Pipe-traversal (Fig. 2.3E, 2.9E)	17	42
Roller (Fig. 2.3F, 2.9F)	0.75	0.37

portion to lift off of the pipe and slide forward. This forward momentum moves the rear portion of the robot forward in a dynamic motion (this dynamic motion is more difficult to accurately model, possibly explaining the larger discrepancy noted in Table 2.1). The roller robot (Fig. 2.9E) travels by alternately rotating its outermost segments while bracing against the floor with the opposing segment (note that this is *not* a wheel; the motors do not rotate through 360°). While the roller terrain in Fig. 2.9E appears uneven, these are artifacts of the backdrop used in the images and do not appreciably affect the robot's movement. A video showing the printer and the robots it produced is available here:

Testing the recyclability of the manufacturing platform involves manually unfolding the crawler robot (i.e. straightening the aluminum wire while leaving the motors attached; automating this step is straightforward) and feeding it back into the printer to produce a robot that locomotes in a different manner (Fig. 2.9E). Both the recycled and non-recycled designs closely resemble their virtual counterparts in terms of morphology and locomotion behavior. Because the approach described here allows robots to be fully recyclable, it could complement methods that employ evolution directly in the physical world. A purely real-world [13] or hybrid approach [24] could be useful for more com-

plex robots, in which there might be a greater discrepancy between simulated and real-world behavior.

2.5.1 Scalability Analysis

The evolved robots described here demonstrate that automatically designing and fabricating a variety of different machines from the same base material is possible. However, the present implementation imposes a few design constraints. The serial topology of the described system requires special care to avoid self-intersection during printing and robot actuation (a step handled automatically by the design software). Also, the choice of wire material is a trade-off between the current printer's ability to bend the wire and the wire's ability to support the emerging structure of the robot; future systems with stronger wire or alternative printing environments (e.g. fluid-bath or micro-gravity) that support the robot as it is being folded could allow much longer robots to be fabricated. Though the individual morphology of the robot being fabricated dictates the size of the self-supporting structure, it is possible to bound the problem by performing a worst-case analysis based on a cantilevered configuration. Equation 2.3 relates the maximum cantilevered length (L) to robot design-length (L^*), wire density (ρ) and diameter (d), motor mass (m_m), spacing between motors (S), and wire yield stress (σ_y):

$$\frac{1}{8}\rho\pi g d^2 L^2 + m_m g \sum_{n=1}^{\lfloor L^*/S \rfloor} (L - nS) - \frac{1}{6}\sigma_y d^3 = 0 \tag{2.3}$$

Solving for L reveals that the longest manufacturable cantilevered design with the current configuration is 1.39 meters with one motor module (i.e. 54 folds

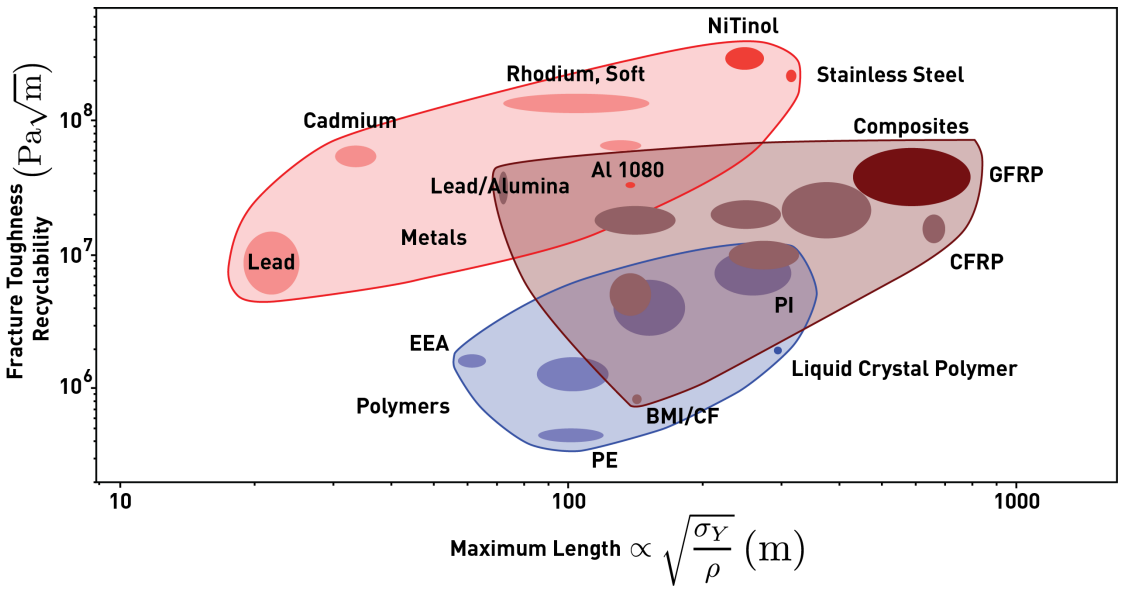

Figure 2.10: Analysis of Candidate Feed Materials

with a 25 mm fold spacing). Robots with a longer overall-length are possible, provided that the system folds them so that the moment arm from the design remains within this maximum cantilevered radius of the bending head.

Extending this analysis to examine the space of possible materials reveals that those with a higher specific strength such as stainless steel or NiTinol wire (Fig. 2.10) could produce robots with twice the cantilevered length of those fabricated with aluminum wire because these relatively lighter and stiffer designs would allow longer sequences of folds to self-support. Because the CPPN representation is generative, it already supports seamlessly scaling to robot designs with higher complexities regardless of the length of the robot enabled by a particular material choice.

The cold-working involved in bending aluminum wire eventually results in its failure through brittle fracture. This failure mode limits the number of

recycling steps to between 10^1 to 10^2 cycles, before the material shatters [74]. The alternative materials shown in Fig. 2.10 also have fracture toughnesses that are superior to aluminum, making them better suited to the cold-working that is involved in recycling.

2.6 Discussion

Robots are currently designed and fabricated manually, leading to high costs and making them time-consuming to produce or adapt to novel scenarios. To address this issue, and enable robots that are simultaneously tailored to an application and inexpensive, roboticists have begun to break from the reliance on manual design and fabrication by using modular design approaches and automated fabrication methods. Recent work with rigid [93, 103, 77] and soft robots [21, 109] has employed a combination of interactive design based on manual composition of modules from a library pre-populated by expert-designers, and a subsequent optimization step to refine these mechanisms based on the application's objectives. In all cases, multiple stages of manual fabrication and assembly are involved to implement the designs. New additive manufacturing techniques have been developed to automatically fabricate complete assembly-free robot mechanisms [71], however, the design phase of these systems still required substantial human decision-making.

In contrast, leveraging ideas from natural assembly processes demonstrates that automatically designing *and* fabricating a variety of different robots is possible. Although the robots shown here have modest functionality, the process and modules used are scalable to permit larger or more complex robots. For exam-

ple, the wireless communication links between the motor modules are bidirectional and transmit sensor data as well as motor control commands. If necessary, a module with more internal volume could accommodate a more powerful processor or additional sensors. Thicker and stiffer wire (selected from Fig. 2.10) combined with more powerful motor modules would allow stronger robots. Provided that the total cantilevered length of the robot adheres to eq. 2.3, there is no limit on the linear length of the robots that can be bent; robots more complex than those shown could be capable of grasping and manipulating objects in the environment by rotating adjacent body segments relative to each other. Additionally, a similar approach could allow multiple robot chains to connect after the printing process: simply bringing the robots into contact with each other would allow them to fold together and interlink, a concept inspired by protein bonding. The non-backdrivable actuators used in this system would provide zero-energy position-holding in the interlocked segments, allowing actuated hinges to behave as permanent latches. This would allow multiple robot chains to combine, providing topological design flexibility: rather than simple chains, combined structures with branching (arm-like) features would be achievable.

Designing and fabricating specialized robots on-demand will allow them to be customized for each application, rather than using more expensive machines that are exhaustively designed to be general-purpose. This advance could enable robots to be rapidly adapted to disaster scenarios or high-risk environments, in which the challenges are not known *a priori*; the robot deployment might take a phased approach in which observer robots assess the scenario and these assessments provide the design objectives for customized robots produced on-demand to address the specific need (e.g. longer legs to surmount an obstacle; a gripper whose shape is customized to reach and grasp an other-

wise inaccessible object). Similarly, the ability to adapt to unknown situations could be valuable in inaccessible or remote areas, including space exploration. This approach points in a new direction, toward *expendable robotics*, in which a printer rapidly fabricates produces customized robots that are consumed by their application, and then recycles them.

2.7 Afterword

The system proposed here was a success, in that it demonstrated the desired functionality. However, there were some lessons learned that would have greatly improved the reliability of the manufacturing system without compromising functionality. These lessons learned came from two major problems that were seen during the implementation of the system. First, the use of plastic deformation of the cantilevered metal wire was an inaccurate and fundamentally limited method of creating a design. These plastic deformations introduced strain hardened portions of material that made the material more brittle and harder to recycle. Second, the decision to cantilever the design out from the printhead produced a large moment in the structure that further constrained the design space. When the printer was in its original configuration (with the feed direction perpendicular to the gravity vector) this moment would cause the wire to twist, introducing inaccuracies at even small sizes. When the printer was eventually hung from the ceiling in order to align the feed direction with the gravity vector, this moment was placed entirely on the site of the bend. Large enough strands of wire, or designs that were not compact, would plastically deform the wire through their own weight, causing inaccuracies in the design.

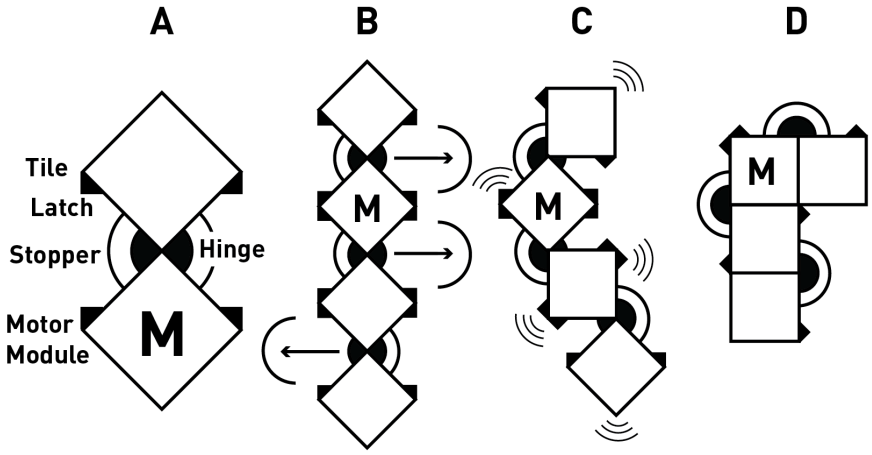

Figure 2.11: Proposed successor ribosome-inspired printer.

A proper design that could address both of these problems would be to take a cue from the ribosome and separate the device that configures the strand of material from the device that produces a conformation from this configuration. In the cell, the ribosome might assemble the strand of amino acids, but it is the cell's environment, the chemical and thermal interactions between the strand and the cytoplasm, that cause the strand to take the shape that makes the protein useful. Therefore, instead of directly bending metal wire, an improved manufacturing system would modify the energy surface of a structure held in an unstable equilibrium, and then perturb this structure to produce the desired conformation as the minimum energy configuration.

As an example, consider a machine that modifies a feed material composed of a string of tiles that are connected to one another with hinges, as shown in Figure 2.11A. If the sides of the tiles touch, there is a latch that locks them into this configuration. There are therefore two configurations that are possible for each hinge, associated with positive and negative rotation of the two tiles relative to one another. This discrete feed material would come fed into the machine

with stoppers that prevent rotation about the hinges. The machine would then selectively remove these stoppers in order to allow only one direction of rotation per hinge, as shown in Figure 2.11B. The modified feed material would then be placed in a space where energy is added to the system- either though the feed material's own actuators or with a shake box, as in Figure 2.11C. These energy perturbations would then provide the energy necessary to bring each tile to touch the sides that allow it to lock, producing the desired configuration. Once the folded automaton has performed its function, the machine can return the feed material into its original state by disengaging the latches and replacing the spacers.

Previous work examined the potential of using one-dimensional strings for manufacturing, mostly focusing on self-actuated strings that were manually designed and assembled[46]. This work was able to show that arbitrary three-dimensional structures could be assembled using space-filling Hamiltonian curves[16], and that simple shapes could be produced using magnetic tiles[46], shake tables, or specially-designed actuators[61].

Given this previous work, and the work already demonstrated with the wire bender, the next step is to close the loop on the manufacturing process and use on-board sensors on the automata to inform the actual performance compared to the expected performance from the physics simulations that are used in the design algorithms. Inertial Measurement Units and external cameras that can estimate simulation parameters like friction and motor response from the actual experimental performance can provide these estimates of actual performance. The simulation running with these improved parameters can then produce a new optimal design for an automaton.

CHAPTER 3

DISTRIBUTED SENSING FOR PERCEPTION IN AERONAUTICAL VEHICLES

This second work examines how an instrumented object, in this case an aircraft, can gain insight into its performance using a large number of sensors distributed across its body. To accomplish this, this work uses a lightweight network that can be embedded in the wing of an aircraft and can provide realtime estimates of the pressure field currently being imposed on the vehicle by the surrounding air. This pressure field not only allows the derivation of critical aerodynamic qualities like lift and drag, but also enables the determination of how the pressure distribution produces these values. Figure 3.1 shows how the distributed sensing work fits into the larger context of a circular manufacturing system. In particular, this work assumes an object already has sensors embedded in it, and distills high-level insights about the state of the object from the large amounts of raw data that these sensors generate.

My contributions for this work included:

1. designing the control boards and the sensor modules and all of the connection hardware for the sensor network
2. writing the firmware that ran on the boards
3. overseeing the integration of the boards into the skin of the airfoil
4. performing the experiments involving the sensor network
5. writing the software for collecting the data and analyzing it
6. formulating the arguments for distributed versus bus-based sensor networks

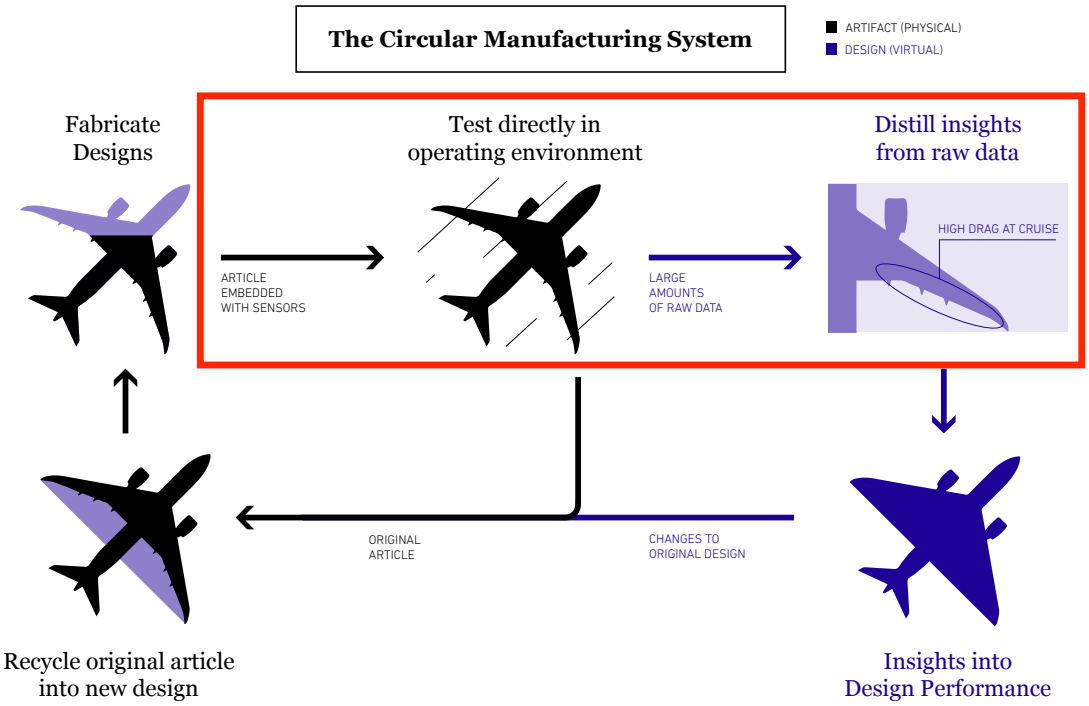

Figure 3.1: The distributed sensing work in the context of a circular manufacturing system.

7. analyzing the collected data in order to produce an estimate of lift from the distributed sensors.

However, I alone did not come up with the Bernoulli algorithm. Credit for that should also go to Nicholas Cramer, my co-author.

In an afterword, I will synthesize the concept of the SPIME with existing visions for the future of aircraft to show how a circular manufacturing system can address some of the challenges inherent to these visions.

3.1 Introduction

The introduction of robotics into mainstream society has the potential to enhance safety while increasing mobility for under-served people through autonomous transportation [57]. There is significant interest and investment into the development of personal air vehicles (PAVs) and various other forms of vertical take-off and landing (VTOL) aircraft, which will need to leverage the current advances in unmanned aerial vehicles to achieve full autonomy [32, 85]. One of the mechanisms to allow for such fully autonomous vehicles is the creation of a self-aware aerial vehicle [5]. The creation of self-aware vehicles would allow for scenarios described in [45] where the combination of knowledge of the vehicle's internal state and external environment allow for graceful degradation of performance as the aircraft ages. This mechanism is especially important for PAVs because there will be no single ideal configuration for the wide variety of missions that these vehicles might undertake and the market for them is predicted to be unable to bear the intensive testing regime that commercial aircraft require to safely transport people [81]. The creation of a self-aware vehicle could help to address these constraints while simultaneously helping to reduce the accident rate for current general aviation vehicles.

The ability of a fully-autonomous aerial vehicle to have a robust estimation of itself and its environment is critical for the vehicle to be able to adapt its behavior, and current research has made progress on this capability with terrestrial robots [24]. This chapter introduces a novel distributed sensing system composed of many imprecise pressure sensors that, when their data are fused together, can provide high-level insight into aerodynamic state, providing the first required level for vehicle self-awareness. The modular, robust, mass-producible

skin introduced here shares many of the same goals as previous work by Buchan et al. [14], where the authors showed the robustness of a properly-designed distributed architecture for an insect robot. Other work has focused on similar modular sensing modules on humanoid robots in order to provide better environmental interaction through tactile sensing. This work extends these lines of enquiry by focusing on the aerodynamic equivalent in the form of air pressure sensors.

In addition to enabling autonomy, aerodynamic state estimation through distributed sensing in aircraft has been performed for gust load alleviation and drag reduction. NASA's SensorCraft model utilized accelerometers, strain gauges, and leading-edge stagnation point sensors, which are typically sets of Pitot tubes or pressure sensors that are used to identify the stagnation point, to perform gust load alleviation in a wind tunnel [94]. Bio-inspired hair sensors developed by the Air Force Research Lab (ARFL) use a carbon nanotube-based mechanism to estimate airflow [75]. Similar artificial hair concepts using silicon-based MEMS techniques can provide lateral line flow sensing in marine robots[37], and the lateral line flow sensing problem space and various solutions bear an intellectual similarity to the proposed method for circulation calculations presented in this paper. The use of potential flow fields and Bayesian filtering can be an effective means for estimating flow conditions [62]. These solutions require either a relatively simple airfoil geometry for which closed-form solutions exist or extensive testing, which might preclude its use for autonomous aerial transportation systems.

This chapter introduces a modular and distributed pressure sensing skin that is capable of real-time aeroelastic state estimation. The distributed MEMS-based

pressure sensors, in conjunction with their localized collection and processing boards, allow for *in situ* information processing and selective loading of the network with useful information instead of unprocessed data. Other work has used similar sensors as a means of aerodynamic state estimation [96, 10] and found them to be satisfactory. This work extends this approach by focusing on mass-manufacturable hardware, flexible architecture, and algorithmic solutions that allow for the distributed information fusion of networks with much higher sensor density.

The mass producibility and flexibility of the system makes it capable of being applied to any arbitrary aircraft design making it especially well suited to the aforementioned PAVs. This flexibility of the algorithms via the distributed knowledge of the aerodynamic pressure field could enable the much sought after *fly by feel* [63], where the primary feedback mechanism is real-time estimates of aerodynamic states. The best implemented example of this mechanism is in biology, where birds like the steppe eagle are hypothesized to have autonomic aeroelastic responses due to pressure changes at high angle of attack without the need for centralized decision making[15]. Such behaviors are thought to be part of what is missing for full-scale robotic autonomy, and are motivation for how the sensor networks proposed here are applicable to flexible aeroelastic aircraft.

The unique requirements of such an instrumentation approach push the limits of conventional wireless sensor networks. Such networks (e.g. [113]) typically operate on much larger scales (10^1-10^3m node separation), with a corresponding increase in network latency (~1 s) and sampling period (~15 min/sample). Recent work, such as TinyNet [91], has focused on networking

protocols that use stateless routing to provide connectivity guarantees that align with this work's focus on scalability and fault-tolerant data transfer.

Further details of the hardware development and estimations techniques follow. Section 3.2.1 describes the hardware and network design decisions that factored into the construction of the distributed sensor network. Section 3.2.2 describes the operation of the experiments that tested the prototype sensor network, and details the component selection and programming of the sensor nodes for this experiment. Section 3.3 examines the collected data and assesses whether it was able to satisfy the criteria described in Section 3.2.1. Section 3.4 describes a few estimation algorithms used on the collected data set, and compares the results of these algorithms to global values measured over the course of the experiment. Finally, Section 3.5 assesses the success of the system's high-level goals and suggests next steps for future systems.

3.2 Methodology

Two parts compose the distributed instrumentation network described in this work: 74 pressure sensors embedded in the skin of a shape-morphing wing, and 22 sensor nodes that collect sensor data, package these data with timekeeping and ID information, and route these packets to a central sink.

The questions this network seeks to answer are as follows:

1. What insights into flow behavior can be deduced from imprecise, low-cost pressure sensors?
2. Can many such sensors sufficiently match existing methods of aerody-

namic state estimation?

3. Can such parameter estimation occur with sufficient speed to be useful in aerodynamic control applications?

3.2.1 Design

The high-level requirements of a system that could answer these questions are therefore as follows:

1. The system shall sample the flow at a sufficient rate to capture interesting flow information
2. The data logs produced by the network shall be able to reconstruct the collected information
3. Hardware designs shall reasonably scale to many hundreds of nodes and thousands of pressure sensors

Given these requirements, a few major design decisions define the configuration of the sensor network. The first and most critical of these decisions is that the network must deliver full sensor data to a central sink rather than attempting *in situ* information fusion. This decision stems from the observation that the test of the sensors as an effective state estimator is independent of their ability to perform this fusion. Despite this decision to use a central sink, the design of the system still retained special focus on the scalability of the system, preferring to make hardware decisions that could feasibly many orders of magnitude more elements even if the firmware used for this particular experiment could not.

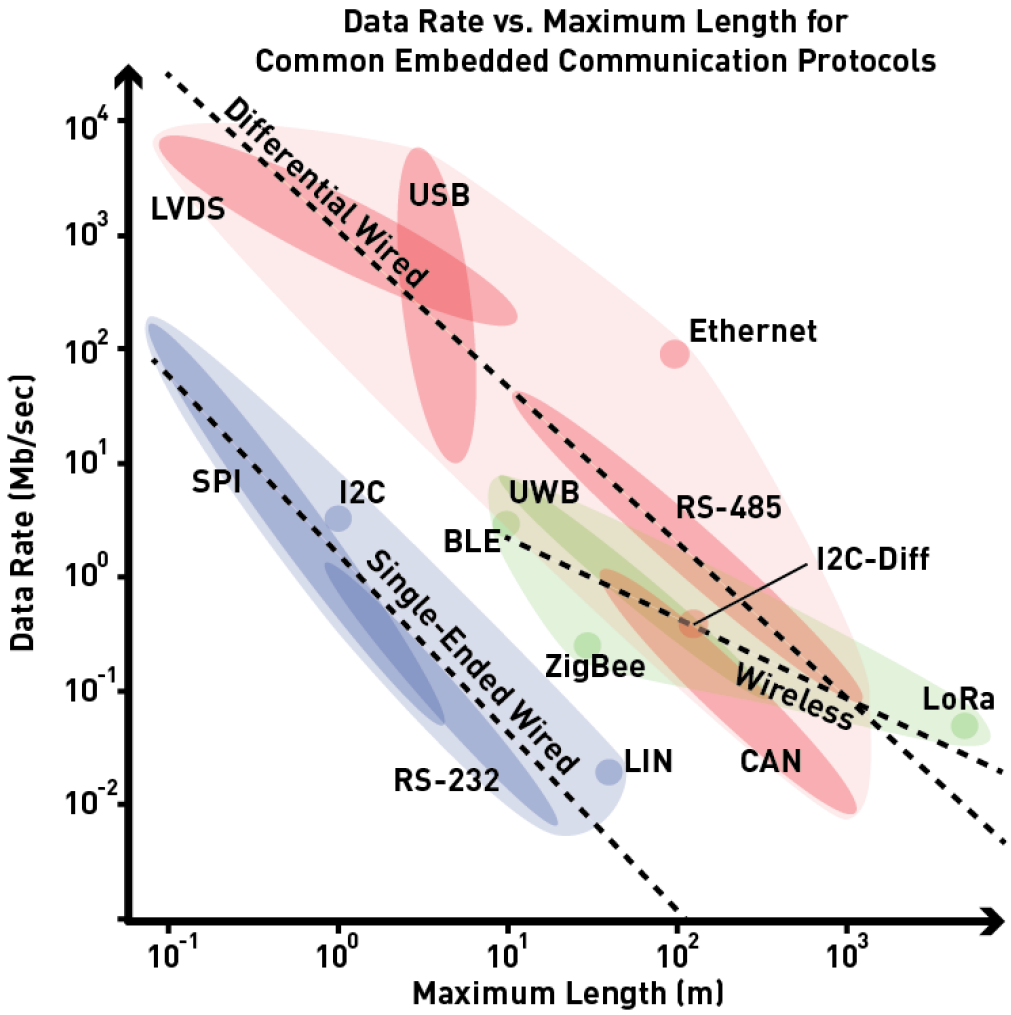

Figure 3.2: Maximum Datarate Versus Maximum Length for Many Common Embedded Communication Standards.

This scalability requirement becomes critical when designing the network architecture. While a conventional system of this size might use a single bus communication standard such as EtherCAT or RS-485 to address each node, such a bus would rapidly become infeasible as the number of nodes becomes very large. Other alternatives exist, including point-to-point serial such as RS-232 or USB, or wireless protocols like Ultrawideband (UWB) and Bluetooth. In order to understand the diversity of choices in selecting a proper communication protocol for embedded applications, Figure 3.2 shows data rates versus maximum length of various common embedded communication standards. The primary source for estimates is Table 14.3 of [53]. Other sources include: LVDS [28], UWB [26], LoRa [27], and differential I2C [87].

While these standards vary in power consumption, topology, and complexity a few key observations come from this chart:

1. Most embedded communication standards follow a distinct negative trend between maximum transmission distance and data rate, due to resonance caused by self-capacitance of the wires.
2. Adding differential signaling can increase the transmission distance of a protocol by two orders of magnitude. This increase comes at the cost of higher complexity in the form of additional components and wires.
3. Higher bandwidth communication is possible, especially in the wireless regime, but not without significant infrastructural/energy requirements.

This chart can also provide insights into the relative efficiencies of bus, hierarchical bus, and mesh networks. Appendix A details the analysis that uses these insights. The result of this study is that the network decided to use wired

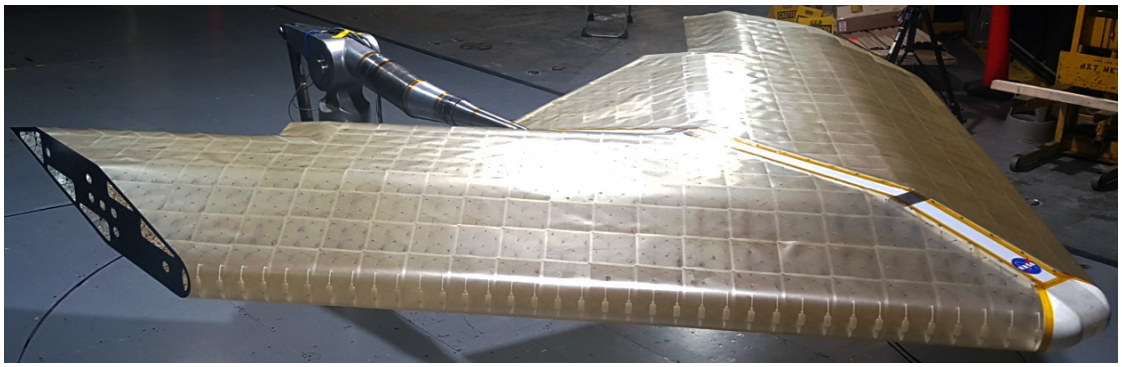

Figure 3.3: MADCAT Aeroelastic Experiment in the NASA Langley Research Center 14x22 wind tunnel.

communication instead of a wireless protocol. Additionally, the choice between wired vs. wireless communication standards is to maximize the simplicity of setup, configuration, and debugging. A wired system also consumes less power and needs less code overhead to run. For instance, a typical UART peripheral requires about 1/100th the power of a BLE system [86], and hard-coded connections between nodes are simpler to set up, since most commercial microcontrollers come with configurable serial interfaces and the routing table is effectively the network topology. With all this said, future work will examine the impacts and performance of wireless technology versus wired for such sensor skins, in particular when these skins use on-board information fusion.

3.2.2 Implementation

The sensor network produced from these design decisions is a part of the MADCAT aeroelastic structure experiment. This experiment investigates a novel, building-block based approach to the creation of aeroelastic structures. It consists of a 14-ft span swept wing constructed from these building-blocks with a

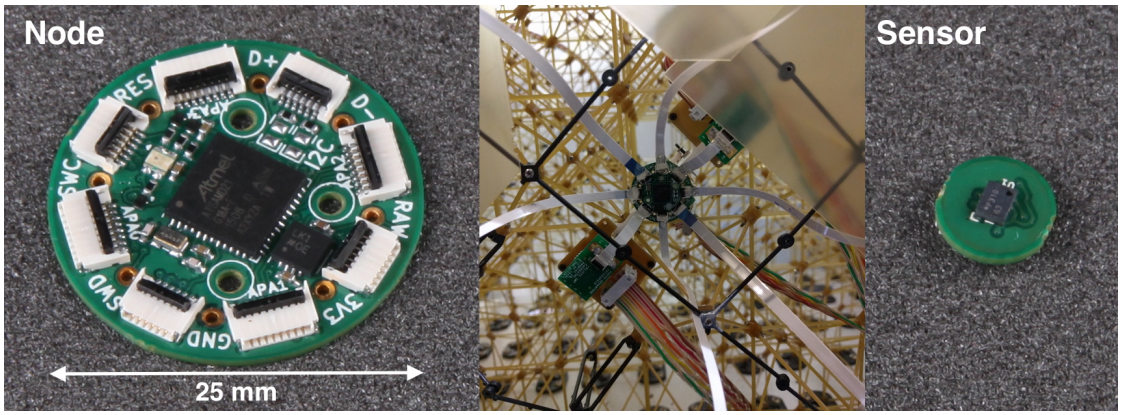

Figure 3.4: The hardware that composes the distributed sensing network.

segmented Ultem 1000 skin. Site of the experiment is the NASA Langley Research Center 14x22 wind tunnel and tests the feasibility of passive structural deformation under aeroelastic load. Figure 3.3 shows this wing as assembled in the wind tunnel.

The sensor network is embedded in one half-span of the MADCAT wing and consists of 22 nodes that collect and route sensor data and 74 sensors that poll the environment for these data. The primary sensor in the network is a commercial off-the-shelf capacitive pressure sensor, the DPS310. Typically used in cellphones as an altimeter, this sensor provides absolute pressure measurements over a range of 300-1200 hPa with a precision of 5 Pa_{RMS} at the maximum datarate of 128 Hz with a current consumption at this rate of 268 μA [55]. The rightmost image in Figure 3.4 shows this sensor along with a mounting board that contains a connector on the opposite side of the board. Each node consists of a SAMD21G18A Cortex M0+ microcontroller, a step-down voltage regulator, a status indication LED, a 32.768 kHz oscillator, and 8 flat flexible cable (FFC) ports that connected nodes with their neighbors or nodes with nearby sensors. Four of the FFC ports connected to separate dedicated UART peripherals on the

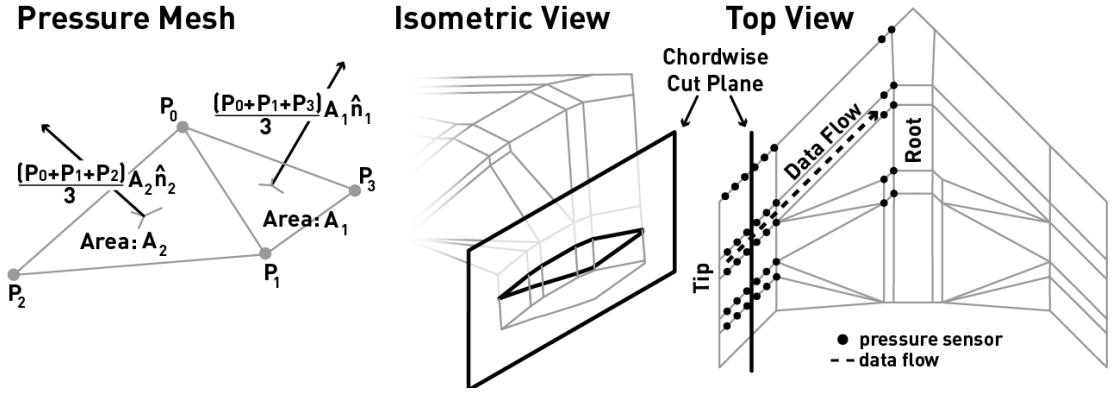

Figure 3.5: LEFT: Illustration of the pressure mesh calculation, for two facets of the mesh formed by the graph of the pressure sensors' locations. The force over the n-th triangular facet is the average pressure of the three sensors at each of the three vertices, multiplied by the area A_n and the unit normal $\hat{n}$ vector. The sum of the vectors from all of the facets, projected along the perpendicular to the free stream flow, produces the total lift.

microcontroller. The leftmost image in Figure 3.4 shows a closeup of this node.

The four UART ports on the node operate at a 1 Mbit data rate with a frame size of 8 bits, no parity, and 1 stop bit with hardware flow control. The other four FFC ports each connect to a single configurable peripheral that could use either the I2C or SPI protocol. Each of these FFC ports routes to a dedicated GPIO pin on the microcontroller that acts as chip select for the sensors connected over the SPI bus. Using this configuration, it is possible to use the same node to connect to four pressure sensors, reducing the number of nodes on the network. Finally, the 32 kHz watch oscillator connects to a 32-bit real time counter hosted as a peripheral on the main microcontroller. This counter increments at a rate of 1024 Hz, and provides node-level timestamping for collected data.

The MADCAT experimental article mounts to a central sting that allows the

automated testing of different angles of attack for the structure. This sting also supplies standard power and ethernet data connections. As a result, the nodes of the network transmit their data to two dedicated central nodes that are identical to the sensor nodes except they are connected to no sensors. These nodes connect over USB to a Startech USB 2.0 over Ethernet extender and transmit the data at a maximum bandwidth of 480 Mbps to a laptop in the control room. A single USB port provides power for the entire sensor network (max. 2.5 W).

Each sensor attaches to a skin panel on the wing using a piece of double-sided tape, and each panel has a small window cut in it to accommodate the pressure port. FFC cables then connect the sensor to the nearest node, which mounts directly onto the substructure, as shown in the center image of Figure 3.4

The sensors can be located along one of eight chord-wise sections, which represent important transition sites in the wing profile, as identified in an XFOIL simulation of the airfoil [31]. These sites are the leading edge (nodes 5 and 6 in Figure 3.6), the transition between the front facets on the top and bottom faces the wing (Nodes 3,4,7, and 8), and the transition between the rear facets on the top and bottom faces of the wing (nodes 1,2,9 and 10). Sensors placed before and after these transitions help mitigate the effect of these transition regions' sharpness on the measured pressure distribution.

Along the span, the network locates most sensors at the tip of the wing, again since simulations indicate that the most interesting aeroelastic behavior are present there. 20 of the sensors that are placed at the root of the wing provide a baseline comparison to the tipward sensors.

In order to transmit information to the sink, the nodes in the network communicate along five spanwise strings of five sensors apiece. Three of these strings connect one of the routing nodes and two connect to the other to reduce the load on any one router. Nodes collect 24-bit pressure data from the sensors that are connected to it, package the data with a timestamp and a node ID, and transmit the packet to a rootward node so that this rootward node can pass this packet along toward the sink. At the sink, a script running on the computer transfers the incoming data to a text file and adds a UTC timestamp every second of operation. This stamp enables correction of the drift in the node RTCs.

Each node connects to its nearest neighbors along the string using a point-to-point serial connection. For connections that are longer than the maximum length of the FFC cables (approximately 25 cm), a twisted-wire ribbon cable extends the range of the UART, as seen in the center image of Figure 3.4. Additionally the leftmost image of Figure 3.5 shows a plan view pointing out the sensor locations on the top surface of the wing. While this image does not show the centerbody nodes that route the collected sensor data to the sink, it does illustrate the general direction of the spanwise data flow from the wing tip to the centerbody with the dashed line.

An equivalent bus over the space of the entire wing requires a minimum datarate of 240 kHz and an overall length of 14 m. Using point-to-point serial connections with the network topology chosen reduces the minimum required datarate to 48 kHz and the maximum required cable length to 1.5 m. From Figure 3.2, it is apparent that the total length and speed requirements of this bus topology put it outside what a single-ended bus could feasibly accom-

plish without investing in additional hardware to upgrade to a differential bus. Using point-to-point connections reduces hardware requirements by reducing both the required communication distance and datarate.

The sensor nodes use a simple state machine and a semaphore on the transmitter resource to route data packets from the tipward nodes to the rootward nodes. Use of interrupts to handle receiving packets and the Direct Memory Access (DMA) Controller for transmitting packets ensured that no polling occurred at the byte level for transmitting data, freeing up CPU time for data collection tasks and maximizing bandwidth.

3.2.3 Experiments

The wind tunnel tests of the MADCAT test article consisted of varying the angle of attack of the wing relative to the flow, while the dynamic pressure of the flow Q ran at a constant 2 pounds per square foot (psf), or approximately 100 Pa. The range of the angle of attack sweep was -4 degrees to 18 degrees in increments of 1 degree. This sweep began and ended with a tare run, where the article proceeded through the same set of angles with the wind off in order to establish a baseline. At each angle of attack, the article held its position until the measured force at the wind-tunnel balance converged, a process that takes between 30 and 60 seconds. We removed these transition periods and took an average of the values in the converged region, in order to find a single pressure value for each angle of attack.

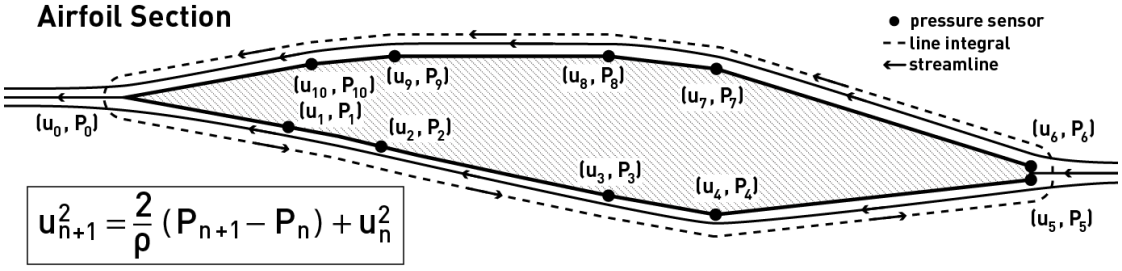

Figure 3.6: A sample airfoil section of the MADCAT wing, with the locations of the pressure sensors and the three closest streamlines. The Incremental Bernoulli Algorithm calculates the lift using a line integral along the dashed contour. The direction of integration is indicated on the contour.

3.3 Results

In total, the testing produced approximately two gigabytes of data. The network successfully transmitted 96% of the data, with three of the sensors on the underside of the wing not being polled due to an address collision on the I2C bus that connected them.

The raw data files that were produced during the experiment consist of unordered lists of data packets containing a node ID, node-time of the collection, and the 24-bit pressure values collected by that node.

Each sensor has a slightly different baseline owing to defects in the manufacturing and calibration process. Removing this baseline variation involves zeroing the value of the measured pressure to the pressure measured during the tare run (when the air in the wind tunnel was off). In addition, all of the sensors exhibit a drift in the measured pressure. This is evidenced by the tare run pressure values taken at the end of the experiment ended being approximately 100

Pa higher than the tare run pressure values taken at the beginning of the experiment. The change in barometric pressure over the course of the morning of the tests accounts for the variation, and inserting the correction from historical data taken at Langley Air Force Base removes this drift [1].

A critical state parameter that can be derived from pressure data is the net lift imparted to the wing by the flow. This value can be calculated in one of two ways: first, by integrating the pressure over the entire surface of the wing, and second, by calculating the circulation around each wing section and integrating those values along the entire span of the wing.

Both methods require knowledge each sensor's physical location. The CAD model of the entire wing provides the rough location of each skin panel that hosts a sensor, and the CAD model of the panel itself provides the precise location of the sensor. Combining these two values together produces a three-dimensional point cloud of values that form the inputs to these two algorithms.

3.3.1 Pressure Mesh Calculation

The direct pressure calculation uses the sensor point cloud to generate a mesh with Delaunay Triangulation [9]. With this mesh, each vertex corresponds to a sensor location with a pressure value, and each facet provides an area and a normal over which the average pressure can be integrated to produce a net force vector. The mean of the three pressure sensors that form the vertices of the facet provide this pressure. The leftmost image of Figure 3.4 shows an example mesh that is formed from four pressure sensors, and shows how the area of the facet, the surface normal, and the average pressure are combined to find a facet

force vector. Summing these force vectors produced the total force vector. The component of this vector normal to the free stream flow is the net lift.

3.3.2 Bernoulli Circulation Calculation

The second method involves partitioning the network into sections with similar spanwise location, producing chord-wise airfoil slices. The sensors in each section sample the pressure at several points on this airfoil slice with sufficient density that they can approximate the pressure distribution over the entire airfoil. By assuming that the airflow was smooth over this foil, Bernoulli's method can calculate the total local stream velocity at each sensor location [62]. Integrating these velocities around the foil produces an estimate for the total circulation around this foil. Integrating the circulations for all the foils over the length of the span produces an estimate for the total lift.

Figure 3.6 shows a sample airfoil with these sensor locations, as well as three of the streamlines for the flow and the direction of the line integral used for calculating the circulation. Combining an initial value for u_0 and P_0 (the free stream flow and ambient pressure, respectively) with the pressure measured at an adjacent center produces the velocity at that point. That is,

$$u_n^2 = \frac{2}{\rho}(P_n - P_0) + u_0^2, \tag{3.1}$$

where n can be either 1 or 10, since those sensors are the closest to the trailing point where value 0 is located. Integration of these velocities then involves per-

forming a line integral of the velocity on the dashed line contour C in Figure 3.6:

$$\oint_C v(s) \cdot \hat{s}\, ds = \sum_{n=0}^{m} \frac{1}{2} (u_{n+1} + u_n) \left[(\vec{x}_{n+1} - \vec{x}_n) \cdot \hat{s} \right] \tag{3.2}$$

The left side of the equation shows the continuous version of the integral, and the right side of the equation shows the linear approximation provided by the sensors. The vector $\vec{x}$ is the physical position of the sensor. By assuming that the the flow is smooth, the velocity at each point is treated as parallel to the surface of the foil, and therefore the dot product $(\vec{x}_{n+1} - \vec{x}_n) \cdot \hat{s}$ is positive on the top of the foil, and negative on the bottom. The location of the stagnation point decided the transition between ”top” and ”bottom”. In order to simplify the calculation, this point occurred between nodes 5 and 6 regardless of the angle of attack of the section.

3.3.3 Centerbody Correction

The measurements from the wind tunnel balance represent the total lift, which is the sum of the left and right wings as well as the centerbody section. Dynamic Tornado [79], a MATLAB implementation of the vortex-lattice method that has been validated for flexible aircraft [23], provided a simulation of the spanwise lift. The full span lift of the simulation matches the balance data from the wind tunnel, enabling the estimation of the contribution of the left wing and center body to the lift force. Subtracting these values from the load balance data therefore produces an estimate of just the right wing that can be compared to the estimated lift values.

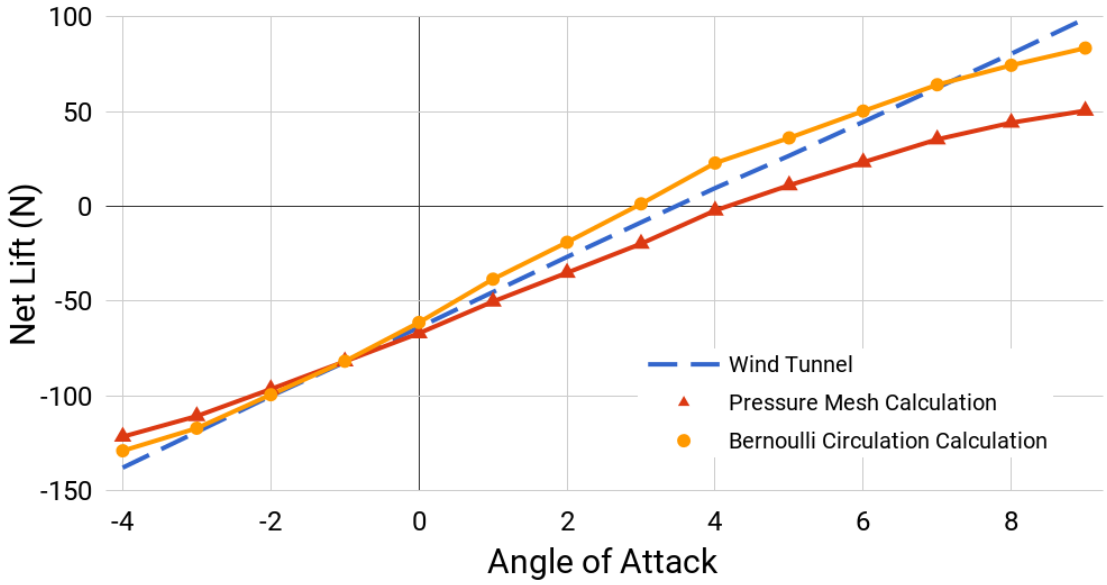

Figure 3.7: Comparison of the lift force measured by the wind tunnel balance and the two different methods for calculating lift from the collected data.

3.4 Discussion

Figure 3.7 shows the two methods alongside the wind tunnel measurements for lift, over an angle of attack range from -4 to 9 degrees. As the figure shows, the pressure mesh calculation consistently underestimates the lift force, for angles of attack greater than 2, while the expected force from the Bernoulli circulation remains close to the measured wind tunnel response. This is likely due to the pressure mesh being sensitive to the undersampled midspan of the wing. While the Bernoulli calculation only interpolates the lift distribution in this midspan, the pressure mesh calculation interpolates the raw pressure values, decreasing the accuracy of the estimation.

3.4.1 Scalability

While the modest number of sensors in this experiment allows the use of a central sink and *post facto* reconstruction, future applications that require real-time analysis or use many times the number of sensors will need to have a strategy for locally collecting and processing data. For the pressure mesh calculation, existing research into power-efficient transmitting in wireless sensor networks has found near-optimal solutions for information fusion over arbitrary networks [51, 66].

For the Bernoulli Circulation calculation, the physics of the algorithm can provide insight into the data flow. Namely, optimal solutions for information fusion will naturally fall along trajectories that are parallel to the free stream flow, since the direction of the line integral and the streamline will be near-parallel. In addition, the Bernoulli Circulation calculation requires external information, in the form of u_0, the free stream flow velocity and P_0, the ambient pressure. These values are usually consistent across the wing, and therefore a single single sensor at the body of the vehicle [111] can provide this measurement for the entire network. Transmitting these values along the trailing edge of the wing to each airfoil section allows the top and bottom nodes of the airfoil to calculate their circulation contributions simultaneously, with the nodes at the leading edge then fusing the result to get a complete estimate of the circulation.

3.5 Conclusions

This work shows that a distributed network consisting of low-cost sensors can accurately extract high-level information about aerodynamic state. It introduces two algorithms for calculating this information, and shows that the version that leverages the physics of the flow produces more accurate results than the naive approach for small angles of attack. This work also shows that the spatial distribution of the sensors provides insight into the lift distribution that would otherwise not be possible to extract from the wind tunnel balance, and uses this information to identify areas to sense in order to improve the accuracy of the system at high angles of attack. Finally, it describes how to extend such approaches to large and/or real-time control networks by intelligently fusing the data at the site of collection.

3.5.1 Future Work

Future work will involve extending this approach in two ways: first, by analyzing the captured data to find algorithms for other flow feature identification tasks, and second, by implementing the algorithms introduced here and found later *in situ*, and, possibly, combine them with actuation for distributed feedback-control of the wing.

The first task involves examining the existing data to find other useful flow features that the network can identify. Some assumptions regarding the flow- for instance, that it was laminar, entirely attached to the foil, and the stagnation point did not move- simplified calculations but could be verified with the

pressure sensors.

The second task involves changing the software present on the network so that nodes can perform more complex operations on collected data than routing it to the sink. Examples include the lift calculation discussed before, where the addition of distributed actuation can optimize wing shape to maximize lift or reduce drag. Accurately finding the chord-wise lift would be a critical parameter in such a feedback control calculation, and, because of the design decisions made in the architecture of the network, adding such capability is a matter of replacing the firmware running on the nodes, and not completely redesigning the system.

3.6 Afterword

The ability to embed sensors in a SPIME is the basis, according to Sterling, of a *Synchronic Society*. In Sterling's words (emphasis his):

> A SYNCHRONIC SOCIETY synchronizes multiple histories. In a SYNCHRONIC SOCIETY, every object worthy of human or machine consideration generates a small history. These histories are not dusty archives locked away on ink and paper. They are informational resources, manipulable in real time. A SYNCHRONIC SOCIETY generates trillions of data-logable, searchable, trackable trajectories: patterns of design, manufacturing, distribution and recycling that are maintained in fine-grained detail. These are the microhistories of people with objects: they are the records of made things in their transition from raw material, through usability, to evanescence, and back

> again to raw material. These informational microhistories are subject to well-nigh endless exploitation.

How does this *Synchronic Society* approach impact how we design aircraft? Currently, the process of iterating on aircraft design involves massive investments in test infrastructure like wind tunnels. This approach makes sense in the current context of aircraft- they spend most of their time operating in a relative static condition (cruise), and the industry produces a relatively small number of aircraft every year.

However, the expected growth of the Urban Air Mobility (UAM) and Unmanned Aerial Vehicle (UAV) sectors will challenge the limits this approach. This challenge is due to two factors: first, the scale of the manufacturing problem, and the environment in which the vehicles will operate. With the former, these smaller, less-expensive aircraft cannot carry the burden of the extensive wind tunnel testing used in conventional aircraft design. Additionally, the need to build a large number of these aircraft will make exhaustively testing for every possible eventuality untenable. With the latter, the operating conditions of these aircraft will likely vary considerably compared to the flight conditions of a current commercial aircraft. The urban environment exhibits flow instabilities, obstacles, and rapid takeoff and landing that have been the most dangerous parts of a commercial aircraft's flight profile.

Instrumentation provides the means for the scale of the first factor to address the challenges in the second. That is, the data produced from these instrumentation systems is one of the trackable trajectories Sterling mentions. It provides a record of the actual operating conditions of the aircraft. Alone, it is interesting, but, in aggregate with the tens of thousands of other aircraft that are also log-

ging their microhistories, it becomes a gestalt from which a safer, more efficient aircraft can derive.

That is, to borrow Sterlings words again:

> The ability to make many small mistakes in a hurry is a vital accomplishment for any society that intends to be sustainable.

The critical first piece of this iterative approach is ensuring that these "small mistakes" are not fatal. This is partly a design problem, partly an autonomy problem. Its a design problem because rapid iteration requires design modifications that test hypotheses derived from the collected data but do not cause the aircraft to be unflyable or unsafe. Its an autonomy problem because, in the event of unpredictable results, robust autonomous systems need to be able to guide the aircraft back to a safe state.

With both of these problems, instrumentation provides the bedrock to improve these decisions. With the former, better data is a necessary,but not sufficient, criterion for better decisions. With the latter, instrumentation provides the robust self-state estimation necessary for autonomy algorithms to operate in a robust manner. A more detailed, more robust sense of self can provide stability by reducing the impact of individual sensor failures and model inaccuracies.

In summary, when responding to the problem of designing a large number of aircraft operate safely in a chaotic, turbulent environment, the synchronic approach is to try everything and see what works. By integrating the production of the aircraft into the operation of the aircraft, it allows the fleet of these SPIME-vehicles to move as a flock through the multivariate design space, rapidly finding the operational optimum. Perhaps an aircraft in this fleet begins with a

design that is the most conservative possible estimate of what is guaranteed to work, but every time a vehicle lands, or there is a lull in activity, or its operating life has ended, it takes the opportunity to review what it and all of its siblings have done since it last self-reconfigure, and it uses the information to improve itself slightly, and perhaps add a small tweak to test a new hypothesis. In this way, the mean performance of all of the aircraft in the fleet begins to approach something that is better than what could have been found using the most exhaustive analysis in the conventional approach.

CHAPTER 4

CELLULAR SOLIDS

This work presents the description and analysis of a novel class of open-cell cellular solids derived from Triply Periodic Minimal Surfaces. This particular class of cellular solids has a few interesting properties: they have locally flat connections, and they can be pre-stiffened in a way that improves their overall performance.

In the context of a circular manufacturing system, cellular solids make an excellent candidate for the materials which compose a SPIME. They can provide high structural performance for their weight, and, if produced from modular units, varying the relative density and material compositon of lattice units can produce structures whose structural performance is tailorable and spatially variant. Finally, they can include reversible connections, which makes the act of recycling materials composed of these structures much easier than with permanent connections. Figure 4.1 shows where cellular solids fit in the larger context of a circular manufacturing system.

My contributions to this work included:

1. discovering a new class of open-cell periodic lattices derived from Triply Periodic Minimal Surfaces
2. analyzing Triply Periodic Minimal Surface (TPMS) lattices using two approaches: a direct stiffness approach and a group-theory based counting rules approach
3. developing a treatment of these lattices that accounts for the collapse modes associated with unit-cell level deformation

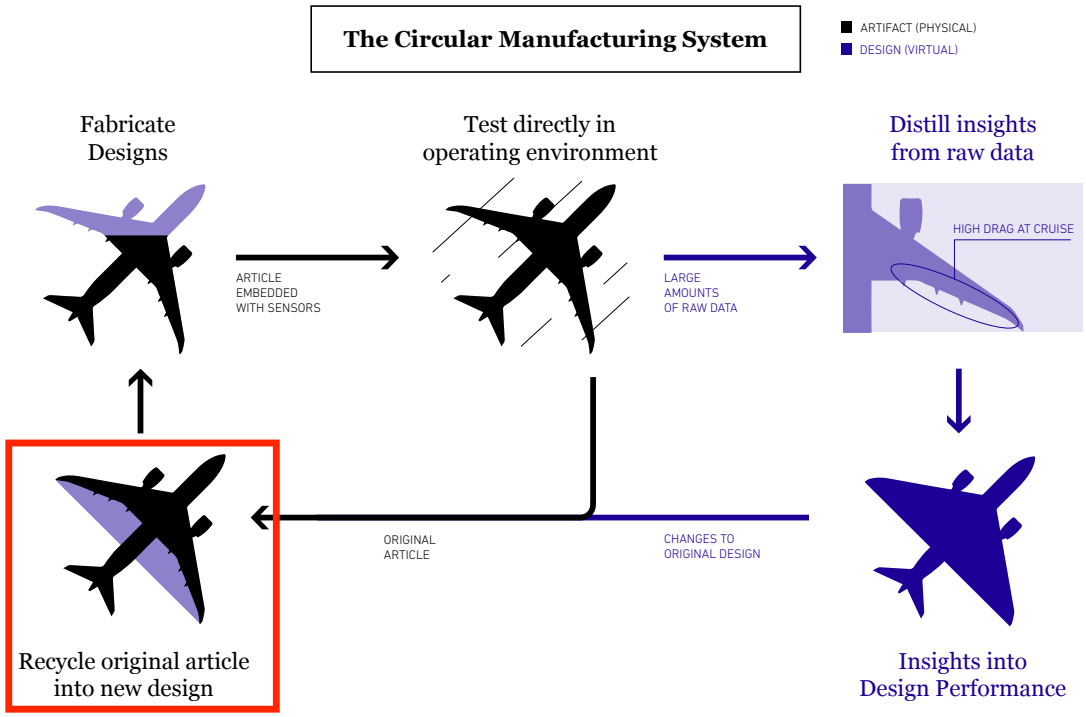

Figure 4.1: The cellular solids work in the context of a circular manufacturing system.

4. writing the program that generates arbitrary lattice geometries from a voxel-based description for construction using additive manufacturing
5. performing experiments indicating that these lattices display unexpected behavior given conventional cellular solids theory

Christine Gregg helped me with the lattice testing, and Kenneth Cheung provided the original idea by asking ”What does a three-dimension warren truss look like?”. The result became the D-Schwarz lattice described here.

4.1 Introduction

Architected cellular solids are a recent approach to materials design that enable the construction of materials with unprecedented stiffness- and strength-to-weight ratios. Researchers have developed architected cellular solids of numerous materials and geometries using lithography processes [114, 56], brazed assemblies of waterjet sheets of titanium[30], three-dimensional woven polymer structures [49], and reversibly-assembled digital cellular composites[18, 44]. Though cellular solids, or foams, refer to any two phase material where one of the phases is a void, architected cellular solids refer to subclass of either open- or closed-cell foams with non-random or ordered microstructure that has been designed or tailored for a specific purpose[39]. Cellular geometries are generated in a variety of ways, the most widespread being the use of three-dimensional honeycombs composed of space-filling polyhedra. Open-cell foams that are derived from these honeycombs use the edges of the space-filling polyhedra to specify their constituent beam networks, while closed-cell foams use the faces of the space-filling polyhedra to specify their constituent walls. The constituent frameworks in the open-cell solids can then be combined to create numerous topologies[115]. Cellular solids have also been generated using Triply Periodic Minimal Surfaces (TPMSes), which display the symmetries of a crystallographic group and can be analyzed using the same techniques developed for convex polyhedral foams. TPMSes have been applied in multiple ways: as "closed" cell foams[4], from graphene structures[89], the Shellular structure[84], and the skeletal graph that defines the gyroid lattice[60].

Ashby developed the conventional theory of mechanical behavior for foams in the mid-twentieth century, in an effort to characterize the behavior of stochas-

tic foams [6], and it is the method for predicting the behavior of both random and ordered foams. The uses the *relative density*, a dimensionless quantity relating the mass of the lattice to the mass of an equivalent volume of material enclosed by the bounding surface of the lattice, and the lattice geometry to predict the beahavior of the various properties of any cellular solid. For instance, the modulus of a cellular solid takes the form of

$$\bar{E} = A\bar{\rho}^{\alpha} \tag{4.1}$$

where $\bar{E}$ is the relative modulus, the modulus of the cellular solid divided by the modulus of the material that constitutes the structural elements of the solid, $\bar{\rho}$ is relative density, and A and α are geometry-dependent constants. A relationship of the same form can be written for strength, fracture toughness, and a variety of other properties. There are several methods and theories for predicting the geometry-dependent terms of these relationships. At the highest level, the mechanical behavior depends on whether the structural elements of the lattice translate the macroscopic forces applied to the entire lattice through bending (moments), or stretching (axially)[29]. A material that translates macroscopic forces axially is called *stretching-dominated* and its mechanical properties scale linearly ($\alpha = 1$) with the relative density. A material that translates macroscopic forces through moments is called *bending-dominated*, and its mechanical properties scale quadratically ($\alpha = 2$) with the relative density.

Application of Maxwell's counting rules for structural rigidity [76] to a pin-jointed version of a lattice provides what is conventionally accepted as a necessary but not sufficient criterion for predicting stretching- versus bending-dominated behavior. If such an application produces an excess of mecha-

nism modes, this theory predicts that this lattice will behave as though it were bending-dominated. If counting produces a surplus of states of self-stress relative to mechanism modes, then the structure has evidence for being stretching-dominated. Guest and Fowler calculated the periodic extension of the scalar counting equation,

$$m - s = nj - b + 2n - 3 \tag{4.2}$$

where $n = 2, 3$ is the space $\mathbb{R}^n$ in which the lattice is embedded.

Since these scalar counting rules provide limited information about how the lattice translates macroscopic applied strains into microscopic deformations, researchers have since extended them using several approaches designed to provide a fuller picture of how the lattice deforms under an applied load. These methods have focused on extending Pellegrino and Calladine's matrix analysis of finite frameworks[88] into periodic lattices and include Hutchinson and Fleck's Bloch-Wave approach[54], Elsayed and Pasini's extension of this approach[36], Vigliotti and Pasini's direct stiffness method[106], and Guest and Fowler's application of group theory to identifying the symmetries of the mechanism and self-stress states[47].

The most notable insight to come from these analyses is the difference between treating the geometry as a *periodic structure* versus treating the geometry as a *lattice material*. In the former treatment, the geometry can undergo any number of deformation modes corresponding to periodic mechanisms in the unit cell. In the latter treatment, the homogenization of the structure, either through application of the Cauchy-Born rule[12] used in Hutchinson-Fleck and Elsayed-Pasini or the multiscale approach used in Vigliotti-Pasini, limits the kinds of deformation modes that the structure can undergo to those that

have length-scales much larger than a unit cell, and therefore limits the kinds of mechanisms that correspond to bending-dominated behavior to those that also produce macroscopic strain. The Kagome lattice[54] was the original example of a lattice exhibiting this requirement. Subsequent examples[36] have likewise been confined to two-dimensions.

This paper introduces two new three-dimensional geometries, the D-Schwarz and P-Schwarz lattices, which use this additional requirement to scaling-behavior in cellular solids in order to produce the lowest known connectivity geometries that display stretching-dominated behavior. In order to illustrate this behavior, we will first introduce the open-cell triply periodic minimal lattice by showing how to derive a framework from the *fundamental patch* of a TPMS. We will then analyze two such lattices derived from the D- and P-Schwarz surfaces using the Guest-Fowler approach to find the finite mechanisms of the periodic structure and Vigliotti-Pasini approach to find the material properties of the lattice material. We will then present experimental modulus test results of the D-Schwarz derived lattice, which confirm that the mechanisms found for the periodic structure do not correspond to macroscopic strains in the lattice material when geometrical effects are taken into account.

4.2 Analysis

Analysis of the TPMS lattices first involves generating these geometries, followed by application of the symmetry-extended counting rules in order to find their behavior as though they were periodic structures. These counting rules provide the mechanisms and self-stress states that conventionally indicate ei-

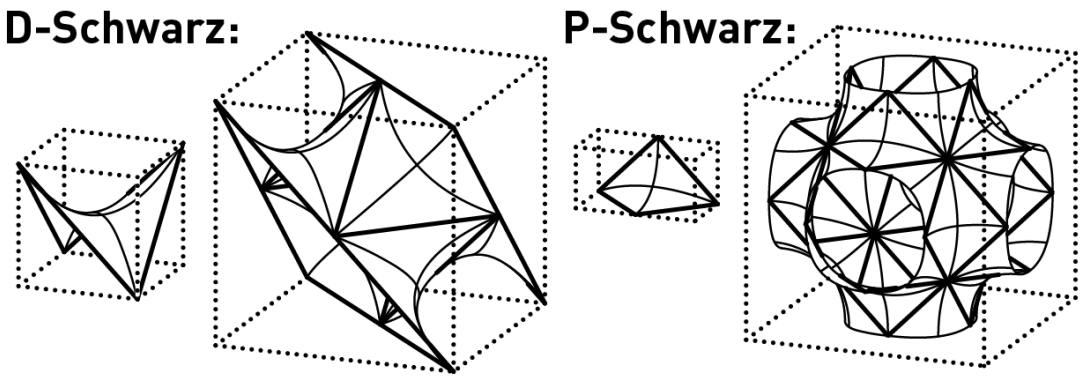

Figure 4.2: The fundamental patch and the unit cell of the TPMSes D-Schwarz and P-Schwarz.

ther stretching- or bending-dominated behavior. The next involves application of the direct stiffness method to the geometries in order to find their behavior as though they were lattice materials Finally, the counting rules and the direct stiffness method combine in order to produce a full picture of the lattice behavior, with the counting rules providing the collapse modes, and the direct stiffness method predicting the macroscopic behavior under these modes.

4.2.1 Generating TPMS Lattices

The first step toward testing these triply periodic minimal surfaces is to develop a general method of deriving them from their constituent surfaces. With convex polyhedral honeycombs like the Octahedral-Tetrahedral or Tridecahedral honeycombs, the edges of the polyhedra that compose the foam provide the bars for open-cell lattices and the faces of the polyhedra provide the surfaces of the closed-cell lattices. Deriving lattices from TPMSes uses a similar approach, called the conjugate surface method. Instead of examining polyhedral cells, the

conjugate surface method uses the *fundamental patch* as the simplest description of the minimal surface. A patch is a polygonal contour over which the Plateau problem can be uniquely solved to find a minimal surface[59]. The straight lines of this polygonal contour then become axes around which this patch can be rotated by π radians to form the full TPMS. From this observation, it is possible to derive the embedded "open-cell" lattice from the boundary of the minimal patch. The straight lines on the boundary of the minimal patch are embedded in the surface, and form the basis of the "open-cell" lattice. Conversely, a minimal patch that contains no straight lines, such as the Gyroid, does not have an "open-cell" lattice. Using this approach, we can derive the lattices for both the D- and P-Schwarz geometries, as shown in Figure 4.2.

4.2.2 Analyzing TPMS Lattices

We analyzed two TPMS lattices and several cellular solid lattices using two approaches: the Guest-Fowler symmetry-extended counting rules and the Vigliotti-Pasini multiscale structural analysis.

The Guest-Fowler approach applies the symmetries of the point group of the lattice to its bars and joints in order to predict the behavior of a pin-jointed periodic structure. It produces a reducible representation of the deformation modes, providing a more detailed insight into the nature of these modes that extends beyond scalar counting. The following equation produces this representation:

$$\Gamma(m) - \Gamma(s) = \Gamma(j) \times \Gamma_T - \Gamma(b) + \Gamma_T \times \Gamma_T - \Gamma_T - \Gamma_R \tag{4.3}$$

Application of Γ to a given object collects the character, $\chi_{object}(S)$, for a given symmetry operation S. This character is the trace of the matrix that that describes the object before and after the application of S. Therefore, $\Gamma(j)$ will collect the number of joints that are undisturbed by the application of each symmetry S in the point group, and $\Gamma(b)$ will collect the number of bars. Additionally Γ_T and Γ_R are the operations corresponding to pure translation and rotation of the unit cell, and are used to compensate for periodicity and solid body transformations.

The result of the above equation produces a reducible representation of the self-stress $\Gamma(s)$ and mechanism $\Gamma(m)$ states. The linear combination of the irreducible representations which produce this reducible representation is provided by the point group that is isomorphic to the space group of the lattice. These irreducible representations correspond to the mechanism (positive) and self-stress (negative) modes in the structure.

Table 4.1: Summary of the three analysis techniques used to predict the behavior of the four studied lattices

Name	m-s	$\Gamma(m) - \Gamma(s)$	Unbiased Lattice			Biased Lattice		
			$\bar{\lambda}_h$	$\bar{\lambda}_d$	$\bar{\lambda}_s$	$\bar{\lambda}_h$	$\bar{\lambda}_d$	$\bar{\lambda}_s$
Octet	-9	$-A_{1g} - E_g - T_{2g} - T_{2u}$	1/3	1/12	1/12	-	-	-
Kelvin	15	$A_{2g} + E_g + T_{1g} + T_{2g} + T_{2u}$	1/3	0	0	-	-	-
D-Schwarz	3	$A_{2u} + T_{1u} - A_{1u}$	1/3	1/12	1/12	0	1/12	0
P-Schwarz	15	$2A_{1g} + 2E_g + 2T_{2g} + T_{1u}$ $+T_{2u} - A_{1u} - E_u$	1/3	1/12	1/12	0	0	0

We applied this approach to four candidate lattices: the two TPMS lattices and two that are well-characterized, the Octet Truss[29] and the Kelvin Foam. Appendix D provides a full description of this application, and Table 4.1 sum-

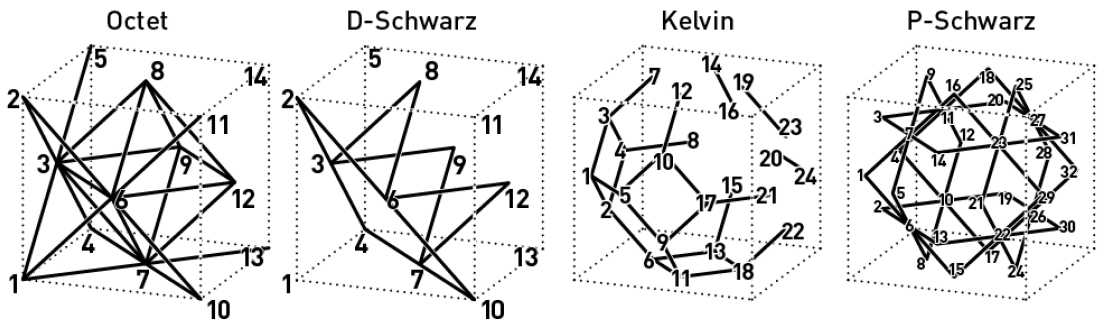

Figure 4.3: Unit Cells of the Octet, D-Schwarz, Kelvin, and P-Schwarz Lattices, with node numbering for the Direct Stiffness Method[106].

marizes the results of this analysis. The results of this analysis show that Octet has exclusively self-stress modes and Kelvin has exclusively mechanism modes. In the case of the D-Schwarz and P-Schwarz lattices, however, there is no indication that any of the self-stress states are capable of stiffening the mechanism modes [48]. Therefore, according to the conventional interpretation of this analysis, the finite mechanisms in this mode will correspond to bending-dominated behavior as a cellular solid.

While counting rules can provide insight into the fundamental mechanism modes of the structure, it is still uncertain whether these mechanism modes correspond to macroscopic behavior of the lattice. Directly estimating the macroscopic stiffness matrix of an infinite cellular solid can determine this correspondance. The Vigliotti-Pasini approach estimates this matrix by applying periodic boundary conditions to a single unit cell using a virtual work approach. This approach requires the generation of a stiffness matrix for a modified unit cell, which is usually produced using finite element modeling, and therefore presents the option of producing pin-jointed or rigid-jointed lattices. We produced custom modeling code that used cylindrical Euler-Bernoulli beam ele-

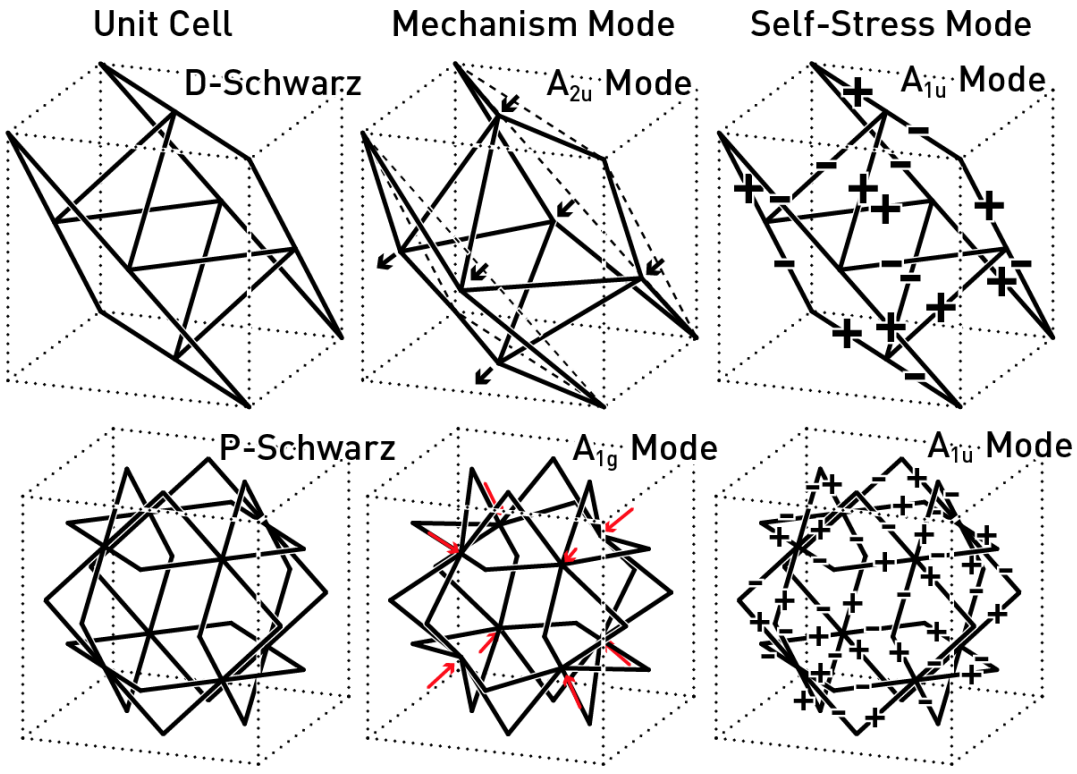

Figure 4.4: The highest-symmetry deformation and self-stress modes for the D-Schwarz and P-Schwarz lattices.

ments to model the bars in the lattice. Figure 4.3 shows the unit cells used in this analysis. In order to correctly identify mechanism modes, we initially generated stiffness matrices that were pin-jointed. That is, the beams carried no bending moments or shears. The results of this analysis are shown in the column labelled "Unbiased Lattice" in Table 4.1.

4.2.3 Combining Theories

Despite the presence of finite mechanisms in the symmetry analysis, the direct stiffness analysis produces no equivalent zero-eigenvalue states for D-Schwarz

and P-Schwarz. This suggests that these lattices are stretching-dominated, despite the presence of these periodic mechanism modes. One explanation for this result is that these mechanism modes are similar to the periodic collapse modes introduced by Hutchinson and Fleck[54], and therefore require that the structure be biased into a state of premature buckling using the primary mechanism mode of the lattice before performing the direct stiffness analysis on the lattice. Figure 4.4 shows two such mechanism modes for the two TPMS lattices: the A_{2u} mode for the D-Schwarz lattice and the A_{1g} mode for P-Schwarz. In the the A_{2u} mode for D-Schwarz, the nodes located on the face of the unit cell translate along along the face diagonal of each respective face either toward or away the origin. In the A_{1g} mode for P-Schwarz, the interior nodes move either toward or away the center of the unit cell. Indeed, after applying these deformations with a pin-jointed lattice, the mechanism modes become present as zero eigenvalues in the stiffness matrix, as shown in the column labelled "Biased Lattice" in Table 4.1.

4.2.4 Experimental Verification

The stretching-dominated behavior of D-Schwarz was experimentally verified using D-Schwarz specimens produced using additive manufacturing (Projet 3600 printer, Visijet M3-X material). To verify relative density scaling, we generated four sets of 10×10×10 lattice volumes generated with 620 μm strut widths at different relative densities, (0.1, 0.05, 0.034, 0.027), and tested these specimens in compression. Full experimental details are available in the Supplementary Materials. The rightmost plot in Figure 4.5 shows the results of this testing, and illustrate that the modulus of the lattice as the relative density is decreased is

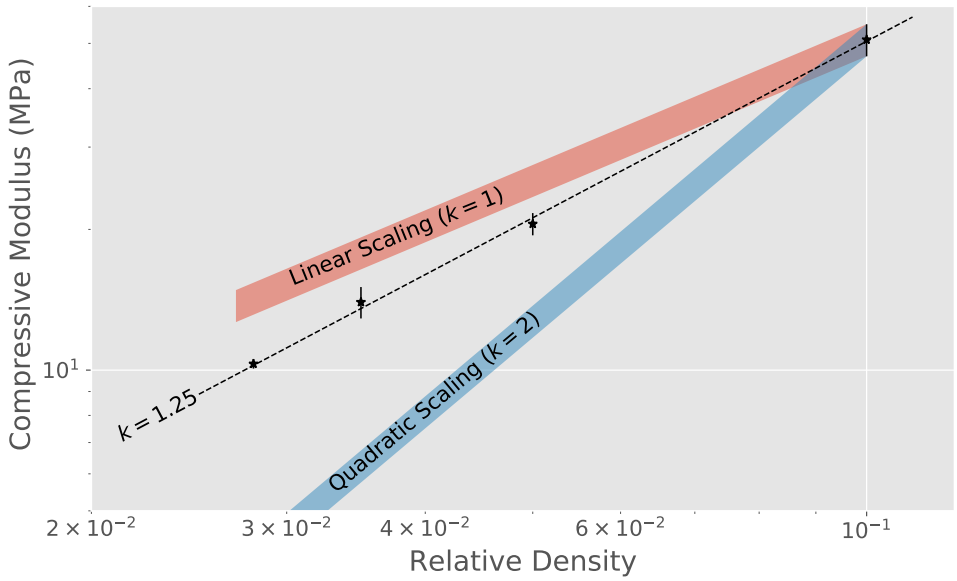

Figure 4.5: Predicted behavior of the pin-jointed D-Schwarz lattice under a constant pre-stress

much closer to linear than quadratic. Appendix C contains a full description of the process for generating the specimens as well as testing them.

4.2.5 Discussion

The results of the experimental verification described above indicate that the D-Schwarz lattice, despite displaying a collapse mechanism, can still exhibit relative-density scaling behavior that is clearly more stretching-dominated than bending.

The closest analog to the behavior observed with D-Schwarz is found in the Bloch-Wave analysis of the Kagome lattice [54]. There, despite having periodic

mechanisms, these mechanisms do not correspond to macroscopic strains and the lattice is therefore considered rigid. It is only when the lattice is biased into the T-T configuration (through the B_1 mechanism mode via counter-rotation of every triangle [47]) that the hydrostatic strain coefficient becomes zero and a finite mechanism mode presents itself. However, this mode is transient; once "geometrical" effects from large displacements are accounted for, the self-stress states of the lattice are able to reinforce the structure, restoring full-rank to the stiffness matrix. The lattices which display this behavior are therefore called *Tensegrity Lattice Materials*, since their performance is dependent on an applied strain [35].

Such *Tensegrity Lattice Materials* have been previously confined to two-dimensions [35]. We propose that the D-Schwarz open-cell lattice represents the first three-dimensional example of this behavior. To verify this proposition, we can examine the stiffness matrix of the pin-jointed lattice in the biased configuration with geometrical effects first ignored and then included.

In order to calculate the comprehensive lattice stiffness in the biased configuration, we propose a *modified Cauchy-Born rule* for the lattices that account for the periodic collapse modes. This rule, instead of relating the position of the nodes of the unit cell with an affine transformation that is proportional to the applied strain, instead displaces the nodes according to the highest-symmetry mechanism- A_{2u} for D-Schwarz and A_{1u} for P-Schwarz. These displacements are detailed in Figure 4.4, and the magnitude of the displacement is sufficient to produce the desired change in the unit cell dimension corresponding to the applied strain. Such modifications to the Cauchy-Born rule are used to explain the behavior of complex crystal structures like perovskites and shape-memory

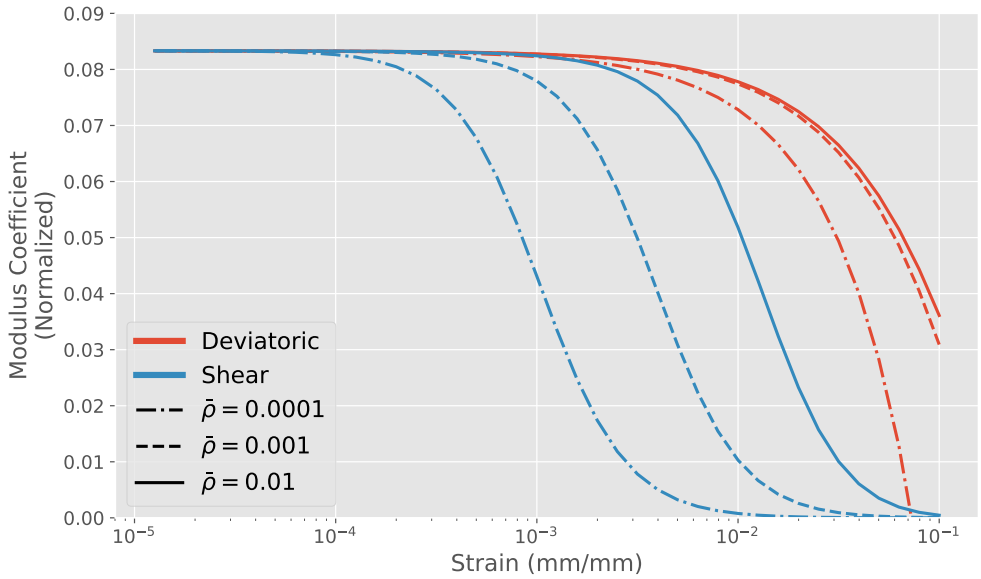

Figure 4.6: Predicted behavior of the pin-jointed D-Schwarz lattice under a constant pre-stress

alloys[112].

In addition, we apply an axial pre-stress to the beams in the unit cell [42] that is configured to correspond to the highest-symmetry self-stress state of the lattice- A_{2u} for D-Schwarz and A_{1u} for P-Schwarz. The details of identifying the self-stress state that informs the force assignments and applying the loads are shown in the Supplementary Materials. The results of this analysis show that the biased pin-jointed D-Schwarz lattice regains a full rank stiffness matrix after accounting for geometrical effects. The impact of this prestress on D-Schwarz' deviatoric and shear stress coefficients is shown in Figure 4.6. For the collapse modes in the biased lattice, there is now a relationship between the relative

stiffness of the lattice $\bar{E}$ and the applied strain ϵ that takes the form

$$\bar{E} = A\frac{\bar{\rho}}{1 + B\frac{\epsilon}{\bar{\rho}}} \tag{4.4}$$

where A is the base stiffness coefficient calculated when the lattice is in unbiased state and B is a correction term. For small values of ϵ, the relative stiffness scales linearly with relative density, but as ϵ grows larger this performance approaches quadratic scaling. Additionally, the strain where this falloff in the shear modulus occurs appears to be related to the square root of the relative density.

4.2.6 Conclusions

We have analyzed a novel class of lattices derived from Triply Periodic Minimal Surfaces, and have shown that, despite the presence of finite mechanism states, these lattices still display stretching-dominated behavior. We have shown that this unexpected behavior is a result of including geometrical effects in the lattice, indicating that these lattices are the first three-dimensional *Tensegrity Lattice Materials*.

The consequences for these kinds of lattices are two-fold. First, unlike fully stretching- or bending-dominated lattices, these lattices exhibit more complex behavior that mixes the qualities of stretching and bending, depending on the amount and type of strain. These qualities enable numerous possibilities for designer lattices that can combine the high-performance of a stretching-dominated lattice at small strains with the energy absorption properties of a bending-dominated lattice at large strains. Second, the performance of these lattices suc-

cessfully demonstrates that conventional counting is not a necessary criterion for predicting the behavior of a lattice- there is an entire space of previously overlooked and undiscovered lattices that can now be fully characterized using the techniques detailed here.

APPENDIX A

BUS SCALABILITY ANALYSIS

This appendix applies the insights from Chapter 3 to analyze the scalability of different network topologies in sensor networks. This analysis focuses on three network topologies: bus, hierarchical bus, and mesh.

1. The simple bus involves a single leader and N followers that are all connected using a common channel. The leader must individually access each follower.
2. The heirarchical bus involves multiple layers of leaders and followers. For the topology examined here consisting of N total nodes, each leader node connects to M followers, and a node can be both a leader and a follower. That is to say, the topology takes the form a regular tree of degree M.
3. The mesh topology involves point-to-point connections between nodes and their nearest neighbors.

Figure A.1 shows examples of the three topologies. There are three parameters with which these different topologies can be compared: bandwidth, length, and latency. Table A.1 lists the values for each parameter for each network type. The following text provides a description of the process for calculating these values, as well as a definition of the variables.

Bandwidth

Given N members of the bus that each transmit b bits, the total data that the network must transmit is bN bits. The bandwidth of the network defined here

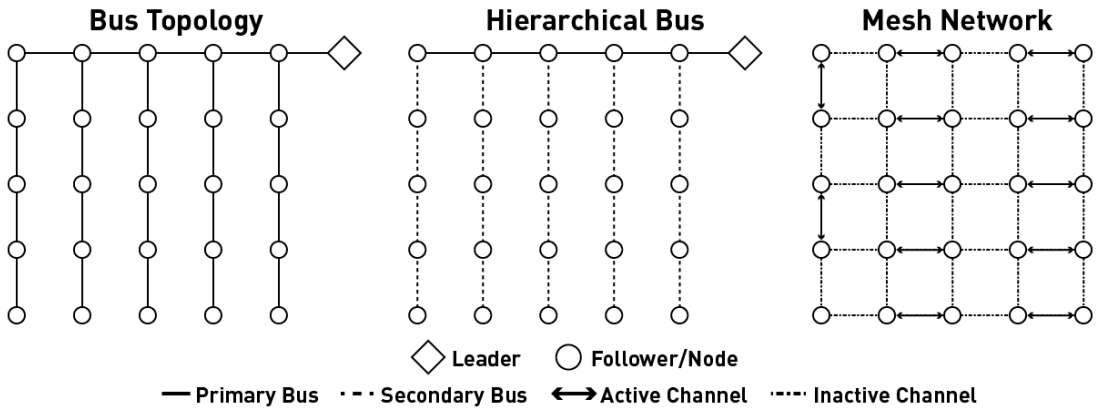

Figure A.1: The three bus topologies analyzed in this appendix.

Table A.1: Scalability Analysis parameters for bus, hierarchical bus, and mesh networks

Network	Bandwidth	Length	Latency
Bus	1	$\sqrt{A(N+1}$	1
Hierarchical Bus	$(N-1)/M$	$\sqrt{A}$	$\log_M(N)$
Mesh Network	$N/2$	$\sqrt{A/N}$	$\sqrt{N}$

is the amount of data the network as a whole can transmit simultaneously, in the form of C channels that represent independent lines of communication. As a result, the network must operate at a rate of bN/C in order to handle all of the data being generated. The term *handling* in this case merely means accessing and receiving, not necessarily transmitting to a central collection point. A key assumption behind this analysis is that the receiver in each communication can fuse the data with its locally collected values and transmit a result onward that does not appreciably add to the data volume. See Section 3.3 of Chapter 3 for an algorithm that displays these properties.

For the bus topology, the leader can communicate with only one follower

at a time. Therefore, the bandwidth is always a single channel ($C = 1$), and the network must operate at a speed of bN in order to collect all of the information for all of the nodes. For the heirarchical bus topology, each node is a leader of M followers. Like the bus topology, these leaders can each communicate with a single follower. Unlike the bus topology, there are $\lceil (N-1)/M \rceil$ nodes with followers that can communicate simultaneously. As a result, there are $C = \lceil (N-1)/M \rceil$ channels in the network, and the network needs to operate at a speed of $\frac{bN}{\lceil (N-1)/M \rceil}$. For large N, this results in a required speed of approximately bM. Finally, the mesh topology allows point-to-point communication between nodes, and the only network collisions that can occur happen when a node tries to communicate with two neighbors at once. As a result, the number of independent channels is $C = N/2$, and therefore the network speed required is just $2b$.

For such an algorithm, the operating speed is also the number of calculations that the leader must perform in order to fuse the data. That is, the value x of the bus speed requirement for each topologies is also the value that determines the time complexity $O(g(x))$ of the algorithm that operates on the data. A topology with a high bandwidth, then, reduces both the required communication speed and the number of computations required at each node, by parallelizing the computation across all of the nodes that are able to receive simultaneously.

Length

As shown in Figure 3.2 in Chapter 3, there is a negative correlation between the overall length of a communication channel and the maximum data rate this

channel can support. Therefore, the length of each of the topologies over a typical distribution of sensors can provide another method for comparing their performance. This comparison assumes that the N followers are distributed evenly over an area A. The dimensions of this area are approximately $\sqrt{A} \times \sqrt{A}$ units.

The length of a traditional bus that reaches all of these sensors has a primary root that is $\sqrt{A}$ units long, followed by $\sqrt{N}$ sub branches that are also of length $\sqrt{A}$ units, for a total length of $\sqrt{A(N+1)}$ units. The length of a hierarchical bus is, in the worst case for the speed of the network, the length of the longest bus. This longest bus must cover a distance on the order of $\sqrt{A}$ in order to reach all of the followers on it. Finally, the distance between any two nodes in the mesh network decreases as the number of nodes increases. Thus, the length of the connections in this topology scale with $\sqrt{A/N}$.

Latency

The latency of a topology is the amount of time necessary to receive data from a specific part of the network. The worst case latency occurs when two nodes that are as far apart as possible communicate with one another.

For the traditional bus, the latency is simply 1, since the leader can communicate with any follower. For the hierarchical bus, the latency is the number of jumps required for the a node at the bottom of the heirarchy to communicate with a node at the top. This is also the number of levels in the hierarchy, which, for a M-regular tree, is $\log_M(N)$. Finally, the worst case latency for the mesh network occurs when two nodes on opposite ends of the network need to communicate with one another. Since the mesh network is approximately

$\sqrt{N}$ nodes wide, then the communication between these maximally-separated nodes requires $\sqrt{N}$ jumps.

Conclusions

These three criteria enable the selection of an optimal bus topology given the spatial extent and the number of nodes in the topology. With a small number of followers clustered close to a leader that needs to be contacted quickly, the traditional bus is likely the best solution. With more sensors distributed over a large area, the hierarchical bus and mesh networks can provide better bandwidth and higher speeds, at the cost of having more network hops and therefore higher latency. In this space, the mesh network trades better bandwidth and faster communication between nearby nodes for worse latency over the entire network compared to the hierarchical bus. The mesh network is therefore a better choice in systems where the collected information is spatially constrained.

With the distributed sensing experiments described in Chapter 3, it is the combination of the bus and a mesh topologies that addresses the particular needs of this system. The bus topology provides the low-latency, close-range communication between the nodes in the network and the pressure sensors nearest them. Once collected, however, the pressure data is spatially constrained. That is, nodes need only reach local consensus regarding their measurements and transmit these filtered data rather than the measurements at full fidelity. As a result, the mesh topology provides the communication that allows nodes to communicate quickly with other nodes in their local neighborhood, while still remaining scalable to large areas and large numbers of nodes.

APPENDIX B

A SCALABLE SERIAL PERIPHERAL INTERFACE (SPI) BUS

The SPI bus is a lightweight protocol for performing serial communication between a microcontroller and many peripherals [53]. Instead of relying on hardware addressing to differentiate between followers on the bus, SPI instead runs a single wire from the leader to each follower's *chip select* port. Activating a particular *chip select* line tells the follower that the leader wants to communicate with it. If there are only a few followers, running an individual *chip select* line for each of them and connecting each line to an individual general purpose input/output (GPIO) pin is sufficient.

However, if there are many followers, a device called a multiplexer can reduce the number of GPIO pins necessary to just 2^N pins. The multiplexer uses the binary combination of the inputs to select one of the outputs. While the multiplexer reduces the number of GPIO pins required to address N followers from N to $\log_2(N)$, there are still N additional wires in the bus. For a large N or a long bus, these additional wires can result in significant additional weight. Given an average distance D of the followers from the leader, the total length of the additional wires is ND.

This appendix describes a method that requires only two additional bus wires and two additional GPIOs, regardless of the number of followers on the bus. Called the *travelling bit* method, this approach uses a deconstructed shift register to allow the serial addressing of each of the followers with minimal infrastructure. This method is distinct from the daisy chaining approach, where each follower's data output connects to the next follower's data input. The latter approach requires components that can explicitly accomodate it, while the

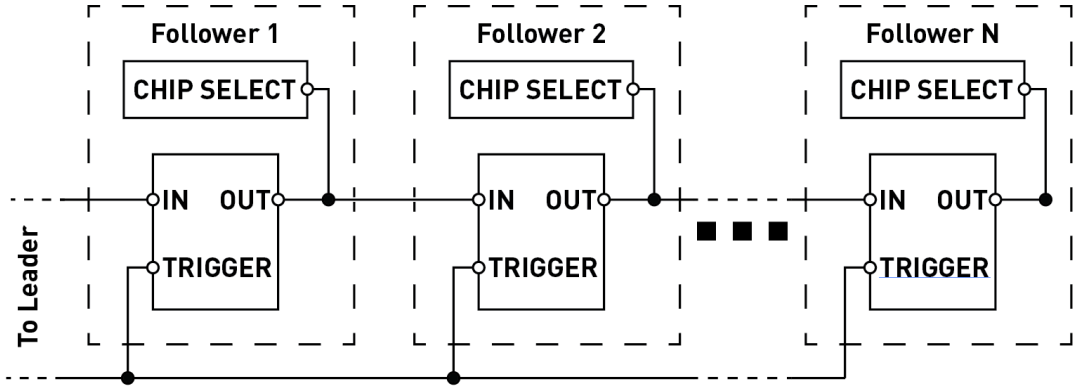

Figure B.1: Electronic schematic of the travelling bit setup.

travelling bit method is follower agnostic- any follower that can implement SPI can implement this protocol.

A shift register is an electronic component composed of a series of circuits called flip-flops that share a common clock. Each flip-flip includes a trigger, an input, and an output. Activating the trigger causes the output value to match that of the input. Until this trigger activates again, the output will hold this initial value, regardless of the input value. A shift register can convert a serial string of values into a parallel array of values (or vice versa) using a daisy chain of flip-flops, where the output of each flip-flop in the series is connected to the next flip-flop's input. A single wire connects all of the triggers, forming the clock line. Activating the clock line causes the values on each of the outputs to shift down one position in the chain. By toggling the clock line and changing the value of the first input, the shift register can set each of the outputs of the flip flops. These outputs form the parallel array of channels.

The *travelling bit* method distributes the components of the shift register onto the SPI bus in order to produce the serial addressing behavior. Each follower on

the SPI bus contains a flip-flop. The output of each flip-flop connects to the chip select of its follower, as well as the input of the next flip-flop in the series. As in the shift register, a single clock line connects all of these flip-flops. Operation of this SPI bus involves the leader entering a single high value into the flip-flop of the first follower, and then moving this bit around by toggling the clock line. Leaving the last output unconnected to any of the other flip flops allows the leader to flush the line by toggling the clock line until the high value leaves the bus. Figure B.1 shows an electronic schematic of this method.

While this method reduces the number of wires, it has one key drawback; the leader can only address each follower in the bus sequentially. In the worst case, the leader needs to address the follower immediately behind the currently activated one, at the very beginning of the bus. The leader would then need to clock the bus N times in order to flush the travelling bit and reintroduce the bit in order to move it one back. If the bit is located further along the bus when it needs to be flushed, then this number is smaller, requiring only $N - m$ clock cycles to activate the $(m - 1)$-th follower when the m-th follower is active. This would mean that there is a period of time when there are two bits on the bus, until the first flushes off the end.

This approach provides the addressing scheme for the distributed Inertial Measurement Unit (IMU) sensor system in the MADCATV0 flight system [17]. There, the truss wing hosts several IMUs distributed along the span of both wings, all of whom must communicate with a BeagleBone Black located in the centerbody that acts as the leader of the bus. Due the tight clearances in the wing, a topology that uses a linear chain, and an application that emphasizes data collection over direct addressing and control, the bus employs the travel-

Figure B.2: The Inertial Measurement Units embedded in the MADCATV0 wing.

ling bit method. Figure B.2 shows the sensors embedded in the MADCATV0 wing (Photo courtesy of Nick Cramer). The method works- the central leader is able to collect information for the several IMUs in each wing at a datarate of 100 Hz.

APPENDIX C

EXPERIMENTAL TESTING OF CELLULAR SOLIDS

Experimental testing involves generating physical representations of the D-Schwarz lattice and testing them under compressive load. In order to do this, custom software generates STL files which are then printed using a multijet additive manufacturing process. This manufacturing and testing process requires preliminary characterization in order to ensure that the observed behavior of the samples accurately reflects the underlying mechanical behavior. Once characterized, D-Schwarz lattices of varying relative density provide the measurements required to estimate the actual scaling behavior of the lattice.

Generating Test Article Lattices Generating a physical artifact for testing from the original lattice involves turning the skeletal graph of the lattice into volumetric components in such a way that minimizes the number of triangles required to specify the test specimen. Splitting the lattice into struts and nodes allows the direct generation of the triangulated surfaces of these volumetric components, as shown in Figure C.1. Because TPMS surfaces are locally flat, simple linear extrusions can represent the nodes of the lattice with two specifiable dimensions: the width of a strut, t, and a *chamfer factor*, c, which alleviates the stress concentration that develops between neighboring bars. The section labelled *Node* of Figure C.1 shows the cross section of such a node. The use of a linear extrusion for the nodes results in bars with a square cross-section at the endpoints; however, the relative orientation of neighboring nodes introduces a 70.53 degree lengthwise twist in the bars. A diagonal along each face of bar allows this twist, producing a pinched cross section at the midpoint of the bar. This pinched cross section has $\approx 75\%$ the area and $\approx 67\%$ the second moment of

the cross section near the node. This smaller area means that the beam is more likely to fail at this midpoint than at a node. The section of Figure C.1 labelled *Bar* shows this twist, as well the effect of the lengthwise of the diagonal on the cross section of the bar at the root and midpoint. The section of Figure C.1 labelled *Microstructure* shows how a bar and two nodes connect to one another.

These two motifs, bar and node, make it possible to construct the entire lattice for a specific relative density $\bar{\rho}$ given two parameters: the strut width t and the chamfer factor c. The result is a lattice with a varying lattice pitch L, defined as the separation between the centroids of any two nodes.

$$\frac{4\,V_{node} + 12\,V_{bar}}{\left(2\sqrt{2}L\right)^3} = \bar{\rho} \tag{C.1}$$

where

$$V_{node} = \frac{3\sqrt{3}}{2} t \left(t + \frac{2}{\sqrt{3}} c\right)^2 \tag{C.2}$$

$$V_{bar} = \left(L - \frac{4}{\sqrt{3}}\left(t + \frac{2}{\sqrt{3}} c\right)\right) V_B t^2 \tag{C.3}$$

and V_B is the volume of the bar with unit length, height, and width.

Constructing a Specimen After generating the geometry for a given strut width, chamfer factor, and relative density, the next step is to fabricate the article for testing. A Projet 3600 Multijet printer fabricates the lattice, which uses an ABS-like photocurable resin called Visijet M3-X and a wax support material called Visijet S300[3]. After printing, a low-temperature over provides the heat necessary to remove the wax support material, leaving the specimen. We

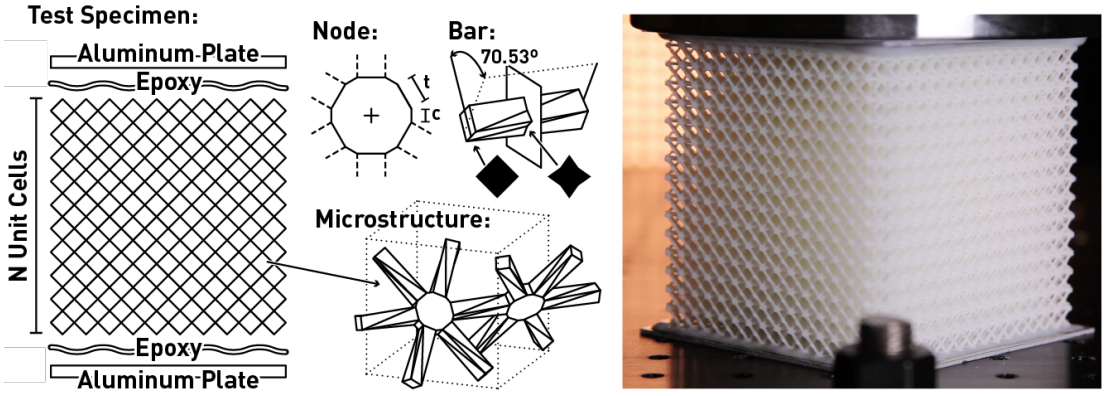

Figure C.1: The procedure for producing D-Schwarz test specimens, and an example specimen being compression tested.

then bond 1/8" 6061 Aluminum sheets onto the top and bottom of these printed specimens, using 3M DP190 Structural Epoxy as the bonding agent, as shown in the *Test Specimen* section of Figure C.1. These plates distribute the load at the interface between the testing apparatus and the specimen evenly across all of the nodes at this interface. The epoxy layer connecting plate to lattice is no thicker than a single node's height ($\approx 1mm$), ensuring that only the nodes are constrained.

Determining Test Article Dimensions This procedure for constructing a specimen introduces two major sources of error that require consideration before testing of the lattice properties. The first is the *edge effect*, the impact of the unconstrained boundary of the sample on the overall performance of the lattice. Increasing the number N of unit cells contained within each dimension of a lattice specimen helps alleviate the impact of this edge effect; for large N, the measured stiffness converges to a value that is the effective modulus of the lattice. The objective of this testing is therefore to find the smallest value of N that

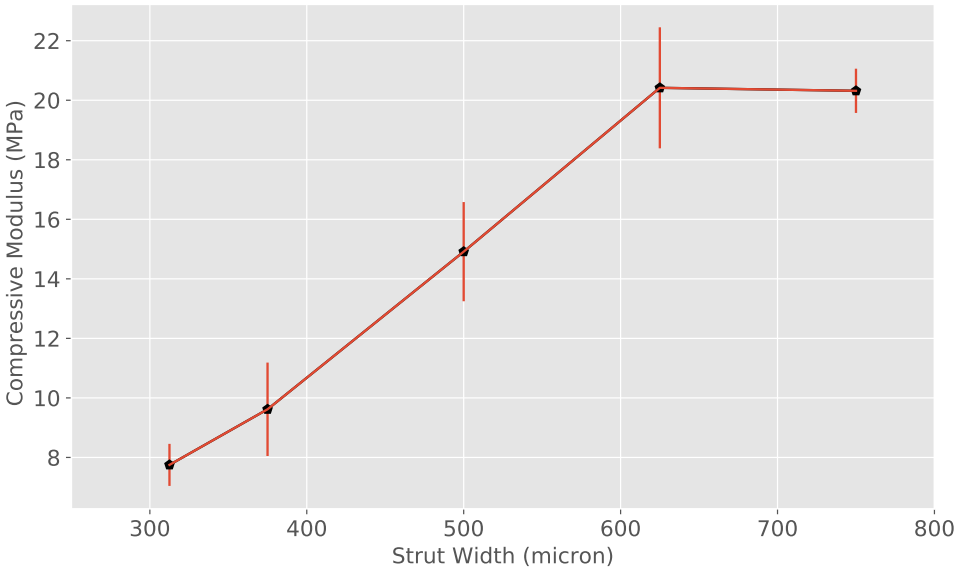

Figure C.2: The compressive modulus of $10 \times 10 \times 10$ unit cell specimens with a relative density of $\bar{\rho} = 0.05$, for increasing strut width t.

produces a modulus that is 90% of this converged modulus. Several $N \times N \times N$ lattices of relative density 0.1, with a minimum N of 4 and a maximum of 18 provide the data necessary to find this value for N. The rightmost image in Figure C.1 shows a picture of the $16 \times 16 \times 16$ unit cell being tested. From this testing an $N \times N \times N$ lattice of dimension $N = 10$ produces behavior that is sufficiently converged.

The second source of error stems from imperfections due to printing. The multijet process that generates the lattices distributes small droplets of plastic material that are then cured using UV radiation. In the Ultra High Definition (UHD) mode used to produce the test specimens, these droplets are roughly $34 \times 34 \times 29\,\mu m$ in size[3]. Several $10 \times 10 \times 10$ lattices with relative density 0.05 and varying strut width t provide the data necessary to find the minimum feature

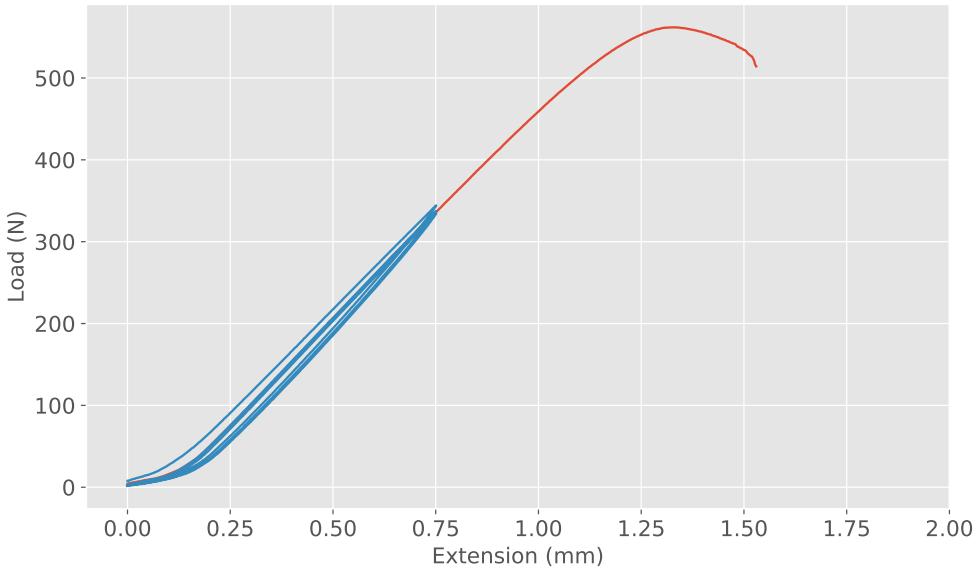

Figure C.3: A sample D-Schwarz lattice undergoing cyclical compression testing between 0 and 1% strain, followed by compression until failure.

size of the lattice where the printed plastic could be treated as a continuum material. The results, shown in Figure C.2, indicate that a strut width of 620 microns is sufficiently large that its mechanical behavior approximates that of a lattice with a larger strut width.

Procedure for Testing Specimens All tests use an Instron 5982 Universal Testing Machine to provide the desired strains and measure the resulting stresses. Beginning with a preload of 50 N, the strain applied to the lattice cycles between 0 and 1% several times at a rate of $0.01\%/s$, in order to get a converged value for the modulus reading. After cycling, the compressive strain increases at the same rate until lattice structural failure. Failure occurs along the (111)-planes of the lattice. Figure C.3 shows the loading behavior of a typical lattice undergoing

compression testing.

APPENDIX D

ANALYSIS OF CELLULAR SOLIDS

The purpose of this appendix is to provide sufficient information to reproduce the results of the Guest-Fowler counting analysis for the identification of mechanism modes in the structure. This section begins with an example analysis for the D-Schwarz lattice in order to show the process, and then provides only the results ($\Gamma(j)$ and $\Gamma(b)$) for the Octet, Kelvin, and P-Schwarz lattices. It then ends with a detailed analysis of the Cuboct lattice.

D.1 D-Schwarz

The orientable D-Schwarz Triply Periodic Surface belongs to space group $Fd\bar{3}m$ (No. 227)[105]. The orientation requirement corresponds to a surface that makes a distinction between the *inner* and *outer* sides of the surface. The embedded lattice analyzed here, however, does not require orientation, and therefore a unit cell of half the dimension and one-eighth the volume is sufficient to describe this simpler geometry. This smaller lattice unit cell belongs to space group $Pn\bar{3}m$ (No. 224)[2]. $Pn\bar{3}m$ contains 10 symmetry operations, which are described in Table D.1. It is isomorphic to the point group O_h with the following mapping.

$$\begin{array}{ccc} O_h & \rightarrow & Pn\bar{3}m \\ \{E, C_3, C_2, C_4, C_4^2, I, S_4, S_6, \sigma_h, \sigma_d\} & \rightarrow & \{E, C_3, C_2, D_4, D_4^2, I, S_4, S_6, g_h, \sigma_d\} \end{array} \tag{D.1}$$

where the operations listed above are the symmetry operations for crystallographic space groups. Note that the lack of a common origin in space group

224 does not affect the mapping, and operations map to their most similar components; screw rotations become rotations, glide reflections become reflections. This isomorphism allows the use of the representations of the unit cell translations and rotations for the point group O_h, as well as the other irreducible representations that compose the group.

The Γ operator collects the character χ of each symmetry operation as it is applied to the joints j and bars b of the lattice. For instance, there are four joints in the unit cell of D-Schwarz, a joint located at the corner of the unit cell, and three joints located at the center of each face of the unit cell. Applying a C_3 operation, corresponding to a $\pi/3$ rotation about a cube diagonal leaves the joint at the corner undisturbed, while disturbing the three joints at the faces of the unit cell. Therefore, the character of the operation $C_3(j)$ is $\chi = 1$. Repeating this process for all of the symmetry operations in the space group produces the value for $\Gamma(j)$.

The resulting value for $\Gamma(j)$ forms a reducible representation of the characters of the joints under the symmetry operations in the space group, $(4, 1, 2, 0, 0, 4, 0, 1, 0, 2)$. This reducible representation decomposes into a linear combination of irreducible representations of the point group O_h[7] by treating the character table for O_h as a matrix and solving for $O_h^{-1}\Gamma(j)$.

Application of a similar process results in the value of $\Gamma(b)$, noting that a bar is considered unchanged even if the joints at its endpoints swap places. Additionally, the values for Γ_T and Γ_R correspond to the translations and rotations of the unit cell as defined by the point group. For O_h, and any space group isomorphic to it, T_{1u} and T_{1g} are the irreducible representations corresponding to these operations.

Table D.1: Symmetry Operations for the Space Group $Pn\bar{3}m$

Name	Description
E	The identity operation, everything stays where it is.
$8C_3$	$2\pi/3$ rotation about the cube diagonal of the unit cell. (2 for each diagonal). In $Pn\bar{3}m$, there is no common center for these operations, instead these axes pass through a joint and are perpendicular to the plane formed by the bars which intersect that joint.
$6C_2$	π rotation about an axis colinear with a bar.
$6D_4$	$\pi/2$ rotation about an axis parallel to a unit cell vector that passes through a point offset $1/4$ unit cell along a second unit cell vector from the origin and $3/4$ in the third direction, followed by translation along that axis $1/2$ unit cell.
$3D_4^2$	The symmetry operation that comes from applying D_4 twice.
I	Inversion centered around a joint.
$6S_4$	$\pi/4$ rotation about an axis parallel to a unit cell vector that passes through a point offset by $1/4$ unit cell from the origin in both of the remaining unit cell vector directions, followed by a reflection through a plane perpendicular to the rotation axis and coincident to a point $1/4$ unit cell along the rotation axis.
$8S_6$	$\pi/6$ rotation about a cube diagonal followed by a reflection through the plane perpendicular to this axis. Like C_3, this operation does not have a common center in the unit cell chosen. These rotation axes will pass through a joint, and the plane will be coincident to a point half the cube diagonal length along this rotation axis from the joint.
$3g_h$	Reflection about a face of the unit cell, followed by a translation along the diagonal of that face for half the length of that diagonal.
$6\sigma_d$	Reflection through a plane coincident to the origin and a face diagonal of the cube.

Using this information, it is possible to construct the vector $\Gamma(m) - \Gamma(s)$ that provides a full description of the mechanism and self-stress modes for a given choice of unit cell. Table D.2 summarizes the calculation that produces this vector. The result $\Gamma(m) - \Gamma(s) = -A_{1u} + A_{2u} + T_{1u}$ indicates that, for this choice of unit cell, there are four mechanism modes, a singly degenerate A_{2u} mode and a triply degenerate T_{1u} mode, as well as one self-stress state A_{1u}. Because the self-stress state does not have the full symmetry of the lattice, it is not capable of blocking the A_{2u} mode and the mode is therefore finite [48].

Collapse Mechanisms from Irreducible Representations Finding the collapse mechanism mode that corresponds to an irreducible represention involves finding the joint displacements that preserve all symmetries with character $\chi = 1$ and breaks all symmetries with $\chi = -1$. For instance, the mode denoted by A_{2u} for space group $Pn\bar{3}m$ breaks the C_2, D_4, I, S_6, and g_h symmetries, and preserves all others. With the D-Schwarz lattice, this mode corresponds to the three joints located at the center of each unit cell face moving along the face diagonal toward the unit cell origin. Figure 4.4 illustrates this mode for the D-Schwarz and P-Schwarz lattices, under the heading *Mechanism Modes*.

Self-Stress States from Irreducible Representations In contrast to the collapse states, the self-stress states in centrosymmetric crystals like D-Schwarz are sets of forces that the bars apply though lengthwise expansion and contraction which create no net movement in the nodes[104]. An edge coloring of a unit cell graph will produce the desired self-stress state provided it satisfies the following two critera:

1. it breaks all symmetries that have character $\chi = -1$ and retains all symmetries that have character $\chi = 1$
2. is anti-symmetric through inversion at each node (that is to say, two bars that swap places when inverted will have opposite colors).

Figure 4.4 also illustrates the highest symmetry self-stress modes for the D-Schwarz and P-Schwarz lattices, in the section labelled *Self-Stress Mode*. Bars labelled with a (+) symbol are those that experience an expansion, while bars labelled with a (−) symbol are those that experience a contraction. Note that P-Schwarz is not centrosymmetric and therefore and therefore the second criterion doesn't apply to all of the nodes.

Other Lattices The other three topologies examined here are the Octet, Kelvin and P-Schwarz lattices. All three belong to space group $Pm3m$ (No. 221), and are isomorphic to the point group O_h. The values for Γ_T and Γ_R and the irreducible respresentations are identical to the analysis of the D-Schwarz lattice, and so the only parts of the analysis that are unique to each geometry are the values of $\Gamma(j)$ and $\Gamma(b)$. As a result, these are the only rows that are included in addition to the result, $\Gamma(m) - \Gamma(s)$. Tables D.3 and D.4 show the calculation for the other three lattices, as well as the reducible representations of $\Gamma(m) - \Gamma(s)$ for each geometry.

D.2 The Cuboctahedral Lattice

The Cuboctahedral (Cuboct) lattice consists of a cubic lattice of vertex-connected octahedra. According to Cheung[18], the lattice displays isostatic behavior ($m - s = 0$ using scalar counting), a relative modulus that scales with $\bar{\rho}^{1.5}$, and

a behavior called *coordinated buckling*, whereby the nodes combine stretching and bending to produce superelastic behavior at large strains[19]. According to Vigliotti and Pasini, the lattice displays linear scaling associated with a stretching dominated lattice[106]. This section applies the analytical techniques described in this thesis to show that the last two qualities are the result of a collapse mode similar to those displayed by the TPMS lattices.

Selection of the Unit Cell The first part of the analysis is the proper selection of the unit cell. Since the cuboctahedral lattice also belongs to the space group *Pm*3*m* (No. 221), the same process for the counting rules applied to the other cubic lattices (Kelvin, P-Schwarz, Octet) also applies with this lattice. Applying the counting rules to the canonical unit cell consisting of a single octahedron produces the null result, with a value of $\Gamma(m) - \Gamma(s)$ that is zero. This result is similar to Guest and Fowler's analysis of the two-dimensional lattice consisting of corner-connected square tiles[47]. There, limiting the unit cell to a single square tile produces a similar null result, and it is only by changing the unit cell, either by increasing its size or moving it, that the mechanism mode becomes apparent. Likewise, increasing the size of the unit cell to include eight octahedra produces a reducible representation that is non-trivial, and, like the TPMS lattices, consists of a mixture of mechanism and self-stress modes. Finally, increasing the unit cell size also shows that the isostatic behavior was an artifact of the choice of unit cell. An isostatic lattice would have $\Gamma(m) - \Gamma(s) = 0$ regardless of the choice of unit cell. Table D.4 shows the result of these counting rules, the first table shows the result of applying the counting rules to a single octahedron, the second shows the result applying these rules to a $2 \times 2 \times 2$ lattice of vertex-connected octahedra. Figure D.1 shows both the $1 \times 1 \times 1$ and $2 \times 2 \times 2$

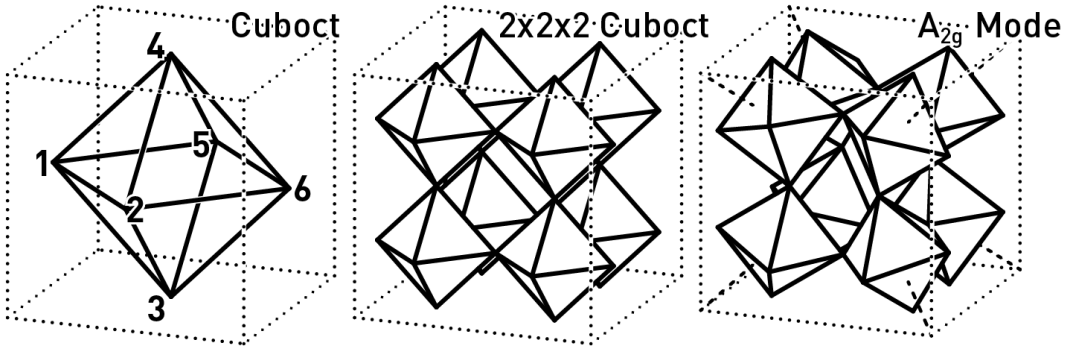

Figure D.1: The cuboct unit cells examined here, and the primary mechanism mode of the periodic lattice.

unit cells.

The reason for this phenomenon, both in the square and cuboct lattices, is that the choice of the unit cell represents the decision to limit the scope of possible modes to those with a corresponding wavevector. For these lattices, there are no modes that occur at the minimum $k = 0$ wavevector regime, and therefore the result is a null representation for $\Gamma(m) - \Gamma(s)$. It is only by expanding the space of possible deformations by adding more degrees of freedom that the modes that define the structures and deformation modes become apparent.

Describing the Modes The highest symmetry mechanism mode has the A_{2u} irreducible representation, and corresponds to solid body rotations of the octahedra about the cube diagonal axis that passes through the octahedron centroid and the unit cell centroid. The rightmost image in Figure D.1 illustrates this deformation mode, as well as the axes around which the octahedra rotate. This mode is auxetic, with rotations corresponding to equal-magnitude reductions of all unit cell dimensions. Qualitative comparison between the expected defor-

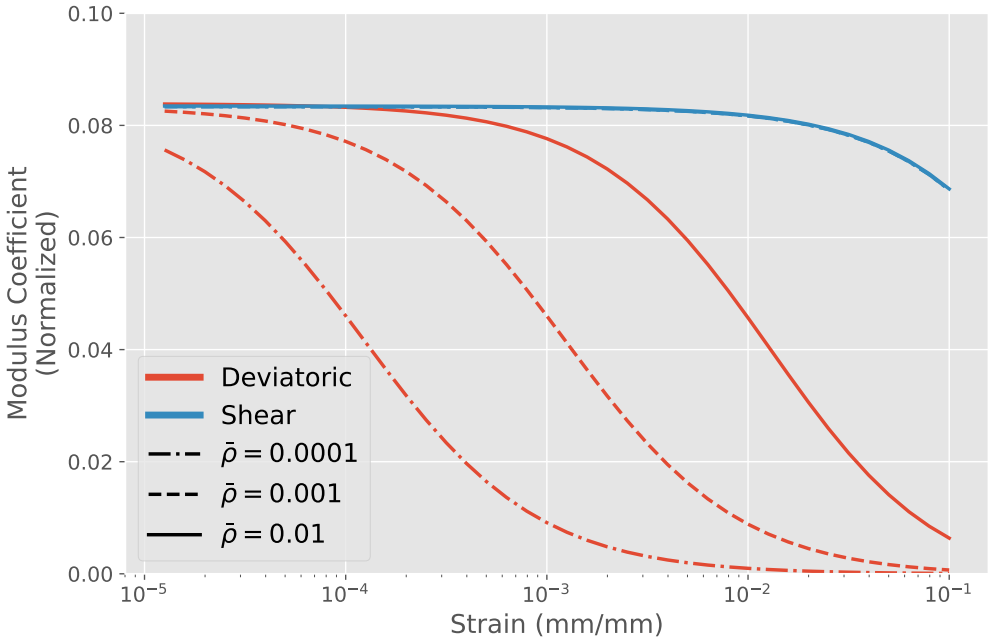

Figure D.2: Effect of strain on the deviatoric and shear stiffness coefficients of the rigid-jointed Cuboct lattice for three different relative densities.

mations due to this mode and the nodal displacements observed in [19] while the lattice is undergoing *coordinated buckling* show that this mode describes the observed deformations.

The highest symmetry self-stress modes have the symmetries of A_{1u} and A_{2u}. The same process used with D-Schwarz provides the force assignments that correspond to these modes. Unlike D-Schwarz, neither of these modes is sufficient to cause the stiffness matrix of the biased pin-jointed lattice to return to full-rank. However incorporating the effect of bending moments returns the biased pin-jointed lattice to full-rank. Incorporating bending moments involves allowing joints to transfer these moments. The same finite element code that

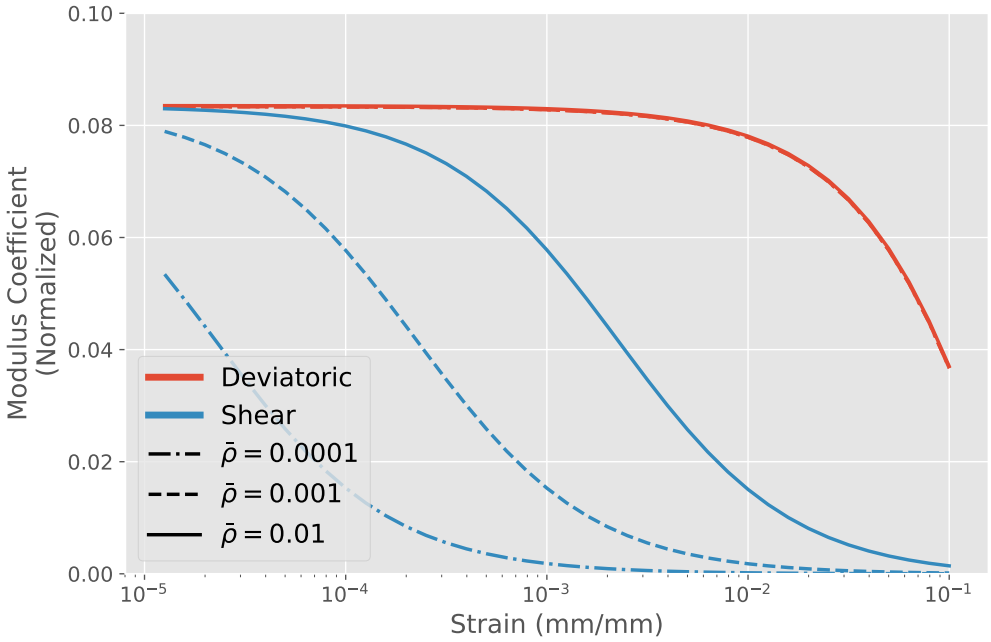

Figure D.3: Effect of strain on the deviatoric and shear stiffness coefficients of the rigid-jointed D-Schwarz lattice for three different relative densities.

produces the pin-jointed lattice can produce the rigid lattice simply by adding these degrees of freedom to the stiffness matrix. The process for doing this is summarized in ¡¿.

Figure D.2 shows the stiffness coefficients of the Cuboct lattice as the unit cell strain increases. The zero eigenvalues of the stiffness matrix for the pin-jointed version of the biased Cuboct lattice correspond to deviatoric strains, so these are the ones that display the collapse mode in this chart. This collapse mode follows a similar trend as the pre-stressed pin-jointed biased D-Schwarz lattice. In this case, the strain that corresponds to the onset of collapse depends linearly on the relative density, rather than on its square root.

This behavior is also present for the rigid-jointed D-Schwarz lattice. Figure D.3 shows such a trend for the shear modulus of the rigid-jointed biased D-Schwarz lattice. Unlike the pre-stressed pin-jointed lattice, the strain corresponding to collapse by shear in the rigid-jointed lattice . This suggests that the pre-stress associated with geometrical effects is a contributor to the near-linear scaling behavior observed in D-Schwarz but not observed in Cuboct.

Table D.2: Full Symmetry-Extended Counting Rule Analysis of the D-Schwarz Lattice

$224 : Pn\bar{3}m$	E	$8C_3$	$3C_2$	$6D_4$	$3D_4^2$	I	$6S_4$	$8S_6$	$3g_h$	$6\sigma_d$	
$\Gamma(j)$	4	1	2	0	0	4	0	1	0	2	$A_{1g} + T_{2g}$
$\times\Gamma_T$	3	0	-1	1	-1	-3	-1	0	1	1	T_{1u}
=	12	0	-2	0	0	-12	0	0	0	2	$A_{2u} + E_u + 2T_{1u} + T_{2u}$
$-\Gamma(b)$	-12	0	-4	0	-4	0	0	0	0	0	$-A_{1g} - E_g - T_{2g} - A_{1u} - E_u - T_{2u}$
=	0	0	-6	0	-4	-12	0	0	0	2	$-A_{1g} - E_g - T_{2g} - A_{1u} + A_{2u} + 2T_{1u}$
Γ_T^2	9	0	1	1	1	9	1	0	1	1	$A_{1g} + E_g + T_{1g} + T_{2g}$
$-\Gamma_T$	-3	0	1	-1	1	3	1	0	-1	-1	$-T_{1u}$
$-\Gamma_R$	-3	0	1	-1	1	-3	-1	0	1	1	$-T_{1g}$
$\Gamma(m) - \Gamma(s)$	3	0	-3	-1	-1	-3	1	0	1	3	$-A_{1u} + A_{2u} + T_{1u}$

Table D.3: Summarized Symmetry-Extended Counting Rule Analysis of the Octet, Kelvin, and P-Schwarz Lattices

Octet											
221 : $Pm\bar{3}m$	E	$8C_3$	$3C_2$	$6C_4$	$3C_4^2$	I	$6S_4$	$8S_6$	$3\sigma_h$	$6\sigma_d$	
$\Gamma(j)$	4	1	2	2	4	4	2	1	4	2	$2A_{1g} + E_g$
$\Gamma(b)$	24	0	4	0	0	0	0	0	8	4	$2A_{1g} + 2E_g + 2T_{2g} + 2T_{1u} + 2T_{2u}$
$\Gamma(m) - \Gamma(s)$	-9	0	-3	1	-1	-3	-1	0	-3	-1	$-A_{1g} - E_g - T_{2g} - T_{2u}$

Kelvin											
221 : $Pm\bar{3}m$	E	$8C_3$	$3C_2$	$6C_4$	$3C_4^2$	I	$6S_4$	$8S_6$	$3\sigma_h$	$6\sigma_d$	
$\Gamma(j)$	12	0	0	0	4	0	0	0	8	0	$A_{1g} + A_{2g} + 2E_g + T_{1u} + T_{2u}$
$\Gamma(b)$	24	0	4	0	0	0	0	0	8	4	$2A_{1g} + 2E_g + 2T_{2g} + 2T_{1u} + 2T_{2u}$
$\Gamma(m) - \Gamma(s)$	15	0	-1	-1	-1	9	1	0	1	-3	$A_{2g} + E_g + 2T_{1g} + T_{2g} + T_{2u}$

P-Schwarz											
221 : $Pm\bar{3}m$	E	$8C_3$	$3C_2$	$6C_4$	$3C_4^2$	I	$6S_4$	$8S_6$	$3\sigma_h$	$6\sigma_d$	
$\Gamma(j)$	20	2	0	0	4	0	0	0	8	4	$2A_{1g} + A_{2g} + 2E_g + T_{2g} + A_{2u} + 2T_{1u} + T_{2u}$
$\Gamma(b)$	48	0	0	0	0	0	0	0	0	0	$A_{1g} + A_{2g} + 2E_g + 3T_{1g} + 3T_{2g}$ $A_{1u} + A_{2u} + 2E_u + 3T_{1u} + 3T_{2u}$
$\Gamma(m) - \Gamma(s)$	15	0	3	-1	-1	9	1	0	9	5	$2A_{1g} + 2E_g + 2T_{2g} - A_{1u} - E_u + T_{1u} + T_{2u}$

Table D.4: Summarized Symmetry-Extended Counting Rule Analysis of Cuboct lattice for two different size unit cells

Cuboct 1×1×1											
221 : $Pm3m$	E	$8C_3$	$3C_2$	$6C_4$	$3C_4^2$	I	$6S_4$	$8S_6$	$3\sigma_h$	$6\sigma_d$	
$\Gamma(j)$	3	0	1	1	3	3	1	0	3	1	$A_{1g} + E_g$
$\Gamma(b)$	12	0	2	0	0	0	0	0	4	2	$A_{1g} + E_g + T_{2g} + T_{1u} + T_{2u}$
$\Gamma(m) - \Gamma(s)$	0	0	0	0	0	0	0	0	0	0	0

Cuboct 2×2×2											
221 : $Pm3m$	E	$8C_3$	$3C_2$	$6C_4$	$3C_4^2$	I	$6S_4$	$8S_6$	$3\sigma_h$	$6\sigma_d$	
$\Gamma(j)$	24	0	0	0	0	0	0	0	8	4	$A_{1g} + A_{2g} + 2E_g + T_{1u} + T_{2u}$
$\Gamma(b)$	96	0	0	0	0	0	0	0	0	8	$2A_{1g} + 2E_g + 2T_{2g} + 2T_{1u} + 2T_{2u}$
$\Gamma(m) - \Gamma(s)$	-21	0	-1	-1	3	9	1	0	9	-3	$A_{2g} + E_g - T_{1g} - 2T_{2g} - A_{1u} - A_{2u} - 2E_u - 2T_{1u} - T_{2u}$

BIBLIOGRAPHY

[1] Noaa national centers for environmental information local climatological data station, langley air force base, va us. https://www.ncdc.noaa.gov/cdo-web/datasets/LCD/stations/WBAN:13702/detail. Accessed: 2018-02-20.

[2] *International Tables for Crystallography*, volume A1, chapter 2.3. International Union of Crystallography, 2011.

[3] 3D Systems. *MultiJet Plastic Printers*, 11 2018.

[4] Diab W Abueidda, Mete Bakir, Rashid K Abu Al-Rub, Jörgen S Bergström, Nahil A Sobh, and Iwona Jasiuk. Mechanical properties of 3d printed polymeric cellular materials with triply periodic minimal surface architectures. *Materials & Design*, 122:255–267, 2017.

[5] Douglas Allaire, David Kordonowy, Marc Lecerf, Laura Mainini, and Karen Willcox. Multifidelity dddas methods with application to a self-aware aerospace vehicle. *Procedia Computer Science*, 29:1182–1192, 2014.

[6] Michael F Ashby and RF Mehl Medalist. The mechanical properties of cellular solids. *Metallurgical Transactions A*, 14(9):1755–1769, 1983.

[7] Peter William Atkins, Mark Sheard Child, and Courtenay Stanley Goss Phillips. *Tables for group theory*, volume 6. Oxford University Press Oxford, 1970.

[8] Joshua E. Auerbach and Josh C. Bongard. Environmental influence on the evolution of morphological complexity in machines. *PLoS Comput Biol*, 10(1):1–17, 01 2014.

[9] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[10] William G Barnwell, Stearns N Heinzen, Charles E Hall Jr, Ndaona Chokani, and David L Raney. Uav flight control using distributed actuation and sensing. 2003.

[11] Josh C Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.

[12] Max Born and Kun Huang. *Dynamical theory of crystal lattices*. Clarendon press, 1954.

[13] Luzius Brodbeck, Simon Hauser, and Fumiya Iida. Morphological evolution of physical robots through model-free phenotype development. *PloS one*, 10(6):e0128444, 2015.

[14] Austin Buchan, Jonathan Bachrach, and Ronald S Fearing. Towards a minimal architecture for a printable, modular, and robust sensing skin. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 33–38. IEEE, 2012.

[15] Anna C Carruthers, Adrian LR Thomas, and Graham K Taylor. Automatic aeroelastic devices in the wings of a steppe eagle aquila nipalensis. *Journal of Experimental Biology*, 210(23):4136–4149, 2007.

[16] K. C. Cheung, E. D. Demaine, J. R. Bachrach, and S. Griffith. Programmable assembly with universally foldable strings (moteins). *IEEE Transactions on Robotics*, 27(4):718–729, Aug 2011.

[17] Kenneth Cheung, Daniel Cellucci, Grace Copplestone, Nick Cramer, Jesse Fusco, Ben Jenett, Joseph Kim, Alexandra Langford, Alex Mazhari, Greenfield Trinh, et al. Development of mission adaptive digital composite aerostructure technologies (madcat). In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 4273, 2017.

[18] Kenneth C Cheung and Neil Gershenfeld. Reversibly assembled cellular composite materials. *science*, page 1240889, 2013.

[19] Kenneth Chun-Wai Cheung. *Digital cellular solids: Reconfigurable composite materials*. PhD thesis, Massachusetts Institute of Technology, 2012.

[20] Jeff Clune and Hod Lipson. Evolving 3D objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution*, 5(4):2–12, 2011.

[21] Fionnuala Connolly, Conor J. Walsh, and Katia Bertoldi. Automatic design of fiber-reinforced soft actuators for trajectory matching. *Proceedings of the National Academy of Sciences*, 2016.

[22] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd

Bickel. Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)*, 32(4):83, 2013.

[23] Nick B Cramer, Sean Shan-Min Swei, Kenneth Cheung, and Mircea Teodorescu. Determination of optimal wing twist pattern for a composite digital wing. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0892. 2018.

[24] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[25] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[26] decaWave. *DWM1000 IEEE 802.14.4-2011 UWB Transceiver Module*, 6 2015. Rev. 1.4.

[27] decaWave. *Low-Power Long Range LoRa Technology Transceiver Module*, 4 2017. Rev. C.

[28] Frank Dehmelt. Performance of lvds with different cables. *Analog Applications*, 2000.

[29] V. S. Deshpande, M. F. Ashby, and N. A. Fleck. Foam topology: Bending versus stretching dominated architectures. *Acta Materialia*, 49(6):1035–1040, 2001.

[30] Liang Dong, Vikram Deshpande, and Haydn Wadley. Mechanical response of ti–6al–4v octet-truss lattice structures. *International Journal of Solids and Structures*, 60:107–124, 2015.

[31] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.

[32] Michael J Duffy, Sean R Wakayama, and Ryan Hupp. A study in reducing the cost of vertical flight with electric propulsion. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 3442, 2017.

[33] N. Eckenstein and M. Yim. Design, principles, and testing of a latching

modular robot connector. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2846–2851, Sept 2014.

[34] Agoston E Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015.

[35] Mostafa SA Elsayed. *Multiscale mechanics and structural design of periodic cellular materials*. PhD thesis, McGill University Library, 2010.

[36] Mostafa SA Elsayed and Damiano Pasini. Analysis of the elastostatic specific stiffness of 2d stretching-dominated lattice materials. *Mechanics of Materials*, 42(7):709–725, 2010.

[37] Zhifang Fan, Jack Chen, Jun Zou, David Bullen, Chang Liu, and Fred Delcomyn. Design and fabrication of artificial lateral line flow sensors. *Journal of micromechanics and microengineering*, 12(5):655, 2002.

[38] S Felton, M Tolley, E Demaine, D Rus, and R Wood. A method for building self-folding machines. *Science*, 345(6197):644–646, 2014.

[39] NA Fleck, VS Deshpande, and MF Ashby. Micro-architectured materials: past, present and future. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2121):2495–2516, 2010.

[40] George E Fox. Origin and evolution of the ribosome. *Cold Spring Harbor perspectives in biology*, 2(9):a003483, 2010.

[41] Jason Gauci and Kenneth O. Stanley. Autonomous evolution of topographic regularities in artificial neural networks. *Neural Comput.*, 22:1860–1898, 2010.

[42] Henri Gavin. Geometric stiffness effects in 2d and 3d frames. 2012.

[43] K. Gilpin, A. Knaian, and D. Rus. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2485–2492. IEEE, 2010.

[44] Christine E Gregg, Joseph H Kim, and Kenneth C Cheung. Ultra-light and scalable composite lattice materials. *Advanced Engineering Materials*, 2018.

[45] Irene M Gregory, Charles Leonard, and Stephen J Scotti. Self-aware vehicles: Mission and performance adaptation to system health. 2016.

[46] S.T. Griffith. *Growing machines*. PhD thesis, Massachusetts Institute of Technology, 2004.

[47] SD Guest and PW Fowler. Symmetry-extended counting rules for periodic frameworks. *Phil. Trans. R. Soc. A*, 372(2008):20120029, 2014.

[48] Simon Guest and Patrick Fowler. Symmetry conditions and finite mechanisms. *Journal of Mechanics of Materials and Structures*, 2(2):293–301, 2007.

[49] Seung Chul Han, Jeong Woo Lee, and Kiju Kang. A New Type of Low Density Material: Shellular. *Advanced Materials*, 27(37):5506–5511, 2015.

[50] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences*, 107(28):12441–12445, 2010.

[51] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, pages 10–pp. IEEE, 2000.

[52] G. S. Hornby, H. Lipson, and J. B. Pollack. Evolution of generative design systems for modular physical robots. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 4, pages 4146–4151 vol.4, 2001.

[53] P Horowitz. The art of electronics–3rd edition/p. *Horowitz W. Hill–NY.: Cambrige University Press, 2015.–1192 p*, 2015.

[54] Robert G Hutchinson and Norman A Fleck. The structural performance of the periodic truss. *Journal of the Mechanics and Physics of Solids*, 54(4):756–782, 2006.

[55] Infineon. *DPS310 - Digital Barometric Pressure Sensor for Portable Devices*, 9 2016. Rev. 1.

[56] Dongchan Jang, Lucas R Meza, Frank Greer, and Julia R Greer. Fabrication and deformation of three-dimensional hollow ceramic nanostructures. *Nature materials*, 12(10):893, 2013.

[57] Nidhi Kalra. Shaping the future of autonomous vehicles. 2016.

[58] Sami Kara, Suphunnika Manmek, and Christoph Herrmann. Global manufacturing and the embodied energy of products. *CIRP annals*, 59(1):29–32, 2010.

[59] Hermann Karcher and Konrad Polthier. Construction of triply periodic minimal surfaces. *Phil. Trans. R. Soc. Lond. A*, 354(1715):2077–2104, 1996.

[60] SN Khaderi, VS Deshpande, and NA Fleck. The stiffness and strength of the gyroid lattice. *International Journal of Solids and Structures*, 51(23-24):3866–3877, 2014.

[61] A.N. Knaian, K.C. Cheung, M.B. Lobovsky, A.J. Oines, P. Schmidt-Neilsen, and N.A. Gershenfeld. The Milli-Motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1447–1453. IEEE, 2012.

[62] Francis D Lagor, Levi D DeVries, K Waychoff, and Derek A Paley. Bio-inspired flow sensing and control: Autonomous rheotaxis using distributed pressure measurements. *Journal of Unmanned System Technology*, 1(3):78–88, 2013.

[63] Les Lee, Ben Dickinson, Jeff Baur, and Greg Reich. Embedded sensors for autonomous air systems, lrir 09rw10cor. Technical report, AIR FORCE RESEARCH LAB EGLIN AFB FL MUNITIONS DIRECTORATE, 2012.

[64] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[65] Zhe Li and Eric Diller. Polymer filament–based in situ microrobot fabrication using magnetic guidance. *International Journal of Advanced Robotic Systems*, 14(1):1729881416682707, 2016.

[66] Stephanie Lindsey and Cauligi S Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings, 2002. IEEE*, volume 3, pages 3–3. IEEE, 2002.

[67] Hod Lipson and Melba Kurman. *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.

[68] Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

[69] Seth Lloyd. Computational capacity of the universe. *Physical Review Letters*, 88(23):237901, 2002.

[70] A. Lyder, R.F.M. Garcia, and K. Stoy. Mechanical design of Odin, an extendable heterogeneous deformable modular robot. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 883–888, Sept 2008.

[71] R. MacCurdy, R. Katzschmann, Youbin Kim, and D. Rus. Printable hydraulics: A method for fabricating robots by 3D co-printing solids and liquids. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3878–3885, May 2016.

[72] Robert MacCurdy, Anthony McNicoll, and Hod Lipson. Bitblox: A printable digital material for electromechanical machines. *International Journal of Robotics Research*, 33(10):1342–1360, 2014.

[73] Yiqi Mao, Kai Yu, Michael S Isakov, Jiangtao Wu, Martin L Dunn, and H Jerry Qi. Sequential self-folding structures by 3d printed digital shape memory polymers. *Scientific reports*, 5, 2015.

[74] JOSEPH MARIN. Cumulative damage and effect of mean strain in low-cycle fatigue of a 2024-t351 aluminum alloy. *Journal of Basic Engineering*, 8:0–1, 1966.

[75] Matthew R Maschmann, Gregory J Ehlert, Benjamin T Dickinson, David M Phillips, Cody W Ray, Greg W Reich, and Jeffery W Baur. Bioinspired carbon nanotube fuzzy fiber hair sensor for air-flow detection. *Advanced Materials*, 26(20):3230–3234, 2014.

[76] J Clerk Maxwell. L. on the calculation of the equilibrium and stiffness of frames. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(182):294–299, 1864.

[77] Ankur Mehta, Joseph DelPreto, and Daniela Rus. Integrated codesign of printable robots. *Journal of Mechanisms and Robotics*, 7(2), 2015.

[78] Ankur M Mehta, Joseph DelPreto, Benjamin Shaya, and Daniela Rus. Cogeneration of mechanical, electrical, and software designs for printable

robots from structural specifications. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2892–2897. IEEE, 2014.

[79] Tomas Melin. A vortex lattice matlab implementation for linear aerodynamic wing applications. 2000.

[80] Shuhei Miyashita, Steven Guitron, Marvin Ludersdorfer, Cynthia R Sung, and Daniela Rus. An untethered miniature origami robot that self-folds, walks, swims, and degrades. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1490–1496. IEEE, 2015.

[81] Mark D Moore. Personal air vehicles: a rural/regional and intra-urban on-demand transportation system. *Journal of the American Institute of Aeronautics and Astronautics (AIAA)*, 2646, 2003.

[82] Jonas Neubert, Abraham P Cantwell, Stephane Constantin, Michael Kalontarov, David Erickson, and Hod Lipson. A robotic module for stochastic fluidic assembly of 3d self-reconfiguring structures. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2479–2484. IEEE, 2010.

[83] Jonas Neubert and Hod Lipson. Soldercubes: a self-soldering self-reconfiguring modular robot system. *Autonomous Robots*, 40(1):139–158, 2016.

[84] Ban Dang Nguyen, Jeong Shik Cho, and Kiju Kang. Optimal design of "shellular", a micro-architectured material with ultralow density. *Materials & Design*, 95:490–500, 2016.

[85] Victoria C Nneji, Alexander Stimpson, Mary Cummings, and Kenneth H Goodrich. Exploring concepts of operations for on-demand passenger air transportation. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 3085, 2017.

[86] Nordic Semiconductor. *nRF52832 Product Specification v1.4*, 10 2017. Rev 1.4.

[87] NXP Semiconductor. *P82B96: Dual bidirectional bus buffer*, 11 2009. Rev. 8.

[88] Sergio Pellegrino and Christopher Reuben Calladine. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures*, 22(4):409–428, 1986.

[89] Zhao Qin, Gang Seob Jung, Min Jeong Kang, and Markus J Buehler. The mechanics and design of a lightweight three-dimensional graphene assembly. *Science advances*, 3(1):e1601536, 2017.

[90] Dan Raviv, Wei Zhao, Carrie McKnelly, Athina Papadopoulou, Achuta Kadambi, Boxin Shi, Shai Hirsch, Daniel Dikovsky, Michael Zyracki, Carlos Olguin, et al. Active printed materials for complex self-evolving deformations. *Scientific reports*, 4, 2014.

[91] Jake Read, Douglas Kogut, Nicholas Selby, and Patrick Wahl. Tinynet: A lightweight networking strategy for robotic control systems. In *Proceedings of ACM Conference*. ACM, 2017.

[92] Sebastian Risi, Daniel Cellucci, and Hod Lipson. Ribosomal robots: Evolved designs inspired by protein folding. In *Proc. of the 15th annual conference on Genetic and evolutionary computation*, pages 263–270. ACM, 2013.

[93] Adriana Schulz, Cynthia Sung, Andrew Spielberg, Wei Zhao, Yu Cheng, Ankur Mehta, Eitan Grinspun, Daniela Rus, and Wojciech Matusik. Interactive robogami: Data-driven design for 3d print and fold robots with ground locomotion. In *SIGGRAPH 2015: Studio*, New York, NY, USA, 2015. ACM.

[94] Robert Scott, David Coulson, Mark Castelluccio, and Jennifer Heeg. Aeroservoelastic wind-tunnel tests of a free-flying, joined-wing sensorcraft model for gust load alleviation. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th AIAA/ASME/AHS Adaptive Structures Conference 13t*, page 1960, 2011.

[95] Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

[96] Wei Shyy, Luis Bernal, Derrick Yeo, and Ella Atkins. Aerodynamic sensing for a fixed wing uas operating at high angles of attack. In *AIAA Atmospheric Flight Mechanics Conference*, page 4416, 2012.

[97] Elin M Sivertsson and Laura S Itzhaki. Protein folding: When ribosomes pick the structure. *Nature chemistry*, 6(5):378–379, 2014.

[98] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162, 2007.

[99] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[100] Kenneth O. Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. 21:63–100, 2004.

[101] Bruce Sterling. The manifesto of january 3, 2000. `http://www.viridiandesign.org/manifesto.html`. Accessed: 2019-03-25.

[102] Bruce Sterling. Shaping things (mediaworks pamphlets). 2005.

[103] Cynthia Sung and Daniela Rus. Foldable joints for foldable robots. *Journal of Mechanisms and Robotics*, 7(2):021012, 2015.

[104] Ellad B Tadmor and Ronald E Miller. *Modeling materials: continuum, atomistic and multiscale techniques*. Cambridge University Press, 2011.

[105] Chaitanya K Ullal, Martin Maldovan, Meinhard Wohlgemuth, Edwin L Thomas, Christopher A White, and Shu Yang. Triply periodic bicontinuous structures through interference lithography: a level-set approach. *JOSA A*, 20(5):948–954, 2003.

[106] Andrea Vigliotti and Damiano Pasini. Stiffness and strength of tridimensional periodic lattices. *Computer methods in applied mechanics and engineering*, 229:27–43, 2012.

[107] S Brett Walker and Jennifer A Lewis. Reactive silver inks for patterning high-conductivity features at mild temperatures. *Journal of the American Chemical Society*, 134(3):1419–1421, 2012.

[108] L. Wang, M. M. Plecnik, and R. S. Fearing. Robotic folding of 2d and 3d structures from a ribbon. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3655–3660, May 2016.

[109] Michael Wehner, Ryan L Truby, Daniel J Fitzgerald, Bobak Mosadegh, George M Whitesides, Jennifer A Lewis, and Robert J Wood. An in-

tegrated design and fabrication strategy for entirely soft, autonomous robots. *Nature*, 536(7617):451–455, 2016.

[110] Sung-Yueh Wu, Chen Yang, Wensyang Hsu, and Liwei Lin. 3d-printed microelectronics for integrated circuitry and passive wireless sensors. *Microsystems & Nanoengineering*, 1, 2015.

[111] Derrick W Yeo. Aerodynamic sensing for autonomous unmanned aircraft systems. 2013.

[112] Giovanni Zanzotto. The cauchy–born hypothesis, nonlinear elasticity and mechanical twinning in crystals. *Acta Crystallographica Section A*, 52(6):839–849, 1996.

[113] Ziran Zhang, Steven D Glaser, Roger C Bales, Martha Conklin, Robert Rice, and Danny G Marks. Technical report: The design and evaluation of a basin-scale wireless sensor network for mountain hydrology. *Water Resources Research*, 53(5):4487–4498, 2017.

[114] Xiaoyu Zheng, Howon Lee, Todd H Weisgraber, Maxim Shusteff, Joshua Deotte, Eric B Duoss, Joshua D Kuntz, Monika M Biener, Qi Ge, Julie A Jackson, Sergei O Kucheyev, Nicholas X Fang, and Christopher M Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344(6190):1373–1377, 2014.

[115] Frank W Zok, Ryan M Latture, and Matthew R Begley. Periodic truss structures. *Journal of the Mechanics and Physics of Solids*, 96:184–203, 2016.